

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY
FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

NÁVRH ŘÍZENÍ VŠESMĚROVÉHO MOBILNÍHO ROBOTU O3-X

DESIGN OF OMNI DIRECTIONAL MOBILE ROBOT (O3-X) CONTROL

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

BC. PETR OLŠA

VEDOUCÍ PRÁCE
SUPERVISOR

ING. RADOMIL MATOUŠEK, PH.D.

Brno, 2010

ZADÁNÍ ZÁVĚREČNÉ PRÁCE

(na místo tohoto listu vložte originál a nebo kopii zadání Vaší práce)

ABSTRAKT

Předmětem této diplomové práce je návrh řízení robotu se třemi všesměrovými koly. Model řízení je navržen pro všesměrovou platformu robotu poháněnou motory od firmy Maxon s inteligentní řídicí jednotkou EPOS.

Návrh řízení obsahuje:

- Navržení souřadných systémů a převodu mezi nimi
- Navržení kinematického modelu soustavy
- Vytvoření knihoven pro řízení a komunikaci s jednotkou EPOS
- Vytvoření aplikace pro simulaci daného problému
- Návrh trajektorie dráhy
- Ověření přesnosti a funkčnosti daného modelu

Navržené řešení bylo založeno na rovnoměrně zrychleném pohybu a bylo uvažováno maximální zrychlení kol, aby nedošlo k jejich prokluzu. U navrženého modelu byla částečně ověřena funkčnost.

KLÍČOVÁ SLOVA

.NET, C#, MCD EPOS, motory Maxon, tříkolový všesměrový podvozek, všesměrová kola, robot, rychlostní řízení, inverzní úloha kinematiky, diskretizace dráhy

ABSTRACT

This thesis deals with design robot control of three-wheeled omni-directional platform. The model of control is designed for omni-directional platform driven by maxon motor with intelligent positioning controller EPOS.

The design control contains:

- installation of the coordinated systems and transformation from one into another
- design kinematical model of system
- creation of classes for control and communications with EPOS
- creation of the simulative program
- planning path mobile robot
- verification that the system is working

The solution was based on continuous accelerated motion and the maximal acceleration of wheels was concerned, so that the slip would be suppressed. The functionality of the model was partly verified.

KEYWORDS

.NET, C#, MCD EPOS, Maxon motor, three-wheeled omnidirectional platform, Omni-directional wheel, velocity control, inverse task of kinematics, path discretization

PODĚKOVÁNÍ

Mé poděkování patří Fakultě strojního inženýrství za vytvoření podmínek a poskytnutí odborného vybavení k realizaci diplomové práce. Jmenovitě bych chtěl poděkovat vedoucímu práce Ing. Radomilovi Matouškovi Ph.D. za odborné vedení. V neposlední řadě patří dík mé rodině za podporu během celého studia.

OBSAH

ABSTRAKT	5
KLÍČOVÁ SLOVA	5
PODĚKOVÁNÍ	7
OBSAH	9
SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ	11
1 Úvod	13
2 Teoretická analýza dané problematiky	14
2.1 Všesměrová mobilita	14
2.2 Všesměrová kola	15
2.2.1 Použitý typ kol	15
2.3 Uspořádání podvozku	16
2.4 Použité motory	16
2.5 Použitá převodovka	18
2.6 Parametry řídicí jednotky EPOS a motoru	20
2.7 Operační módy jednotky EPOS	21
2.8 Postup při komunikaci s jednotkou EPOS	23
3 Praktická část	25
3.1 Zavedení souřadných systémů	25
3.2 Vektory rychlostí kol	26
3.3 Inverzní úloha kinematiky	27
3.4 Aplikace Robot path	29
3.4.1 Hlavní okno aplikace	30
3.4.2 Načtení dráhy z Mobile robot studio	32
3.4.3 Uložení dráhy do XML	33
3.4.4 Nastavení technických parametrů	33
3.4.5 Diagramy nejdůležitějších tříd	34
3.5 Diskretizace dráhy	36
3.6 Relativní a absolutní souřadnice	41
3.7 Bod dráhy	43

4	Ověření funkčnosti zvoleného řešení.....	44
4.1	Příklad dráhy robotu.....	44
4.2	Test aplikace	49
5	Závěr	52
	LITERATURA	53
	SEZNAM PŘÍLOH	55

SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ

α_1	úhel, pod kterým je umístěno kolo 1 vůči ose y Lokálního souřadného systému
α_2	úhel, pod kterým je umístěno kolo 2 vůči ose y Lokálního souřadného systému
α_3	úhel, pod kterým je umístěno kolo 3 vůči ose y Lokálního souřadného systému
α_j	úhel, pod kterým je umístěno kolo j vůči ose y Lokálního souřadného systému
Δt	čas diskretizace
θ	úhel natočení robotu v příslušném souřadném systému
$\theta_G(i)$	úhel natočení robotu v daném bodě Globálního s. s.
$\theta_G(i - 1)$	úhel natočení robotu v předchozím bodě Globálního s. s.
$\theta_{G,i+1}$	úhel natočení robotu v dalším hlavním bodě Globálního s. s.
$\dot{\theta}$	úhlová rychlost těžiště robotu v Globálním s. s.
ω_j	úhlová rychlost kola j
$\omega_j(i)$	úhlová rychlost kola j v daném bodě
ω_{max}	maximální úhlová rychlost kola
ω_{min}	minimální úhlová rychlost kola
N	celkový počet bodů dráhy
R	vzdálenost od těžiště robotu ke středu kola
S	těžiště robotu
a_1	zrychlení kola 1
a_2	zrychlení kola 2
a_3	zrychlení kola 3
a_{max}	maximální zrychlení kola
r	poloměr kola
s_{i+1}	vzdálenost z předchozího bodu do dalšího hlavního bodu
s_i	vzdálenost z předchozího bodu do daného bodu
s	dráha mezi dvěma body
t_f	celkový čas dráhy
v_1	obvodová rychlost kola 1
v_2	obvodová rychlost kola 2
v_3	obvodová rychlost kola 3
v_j	obvodová rychlost kola j
$v_T(i)$	rychlosti těžiště robotu v daném bodě
$v_{T,i+1}$	rychlosti těžiště robotu v dalším hlavním bodě
$v_T(i - 1)$	rychlosti těžiště robotu v předchozím bodě
v_{tran}	čistě translační pohyb všesměrového kola
v_{rot}	čistě rotační pohyb všesměrového kola
$x_G(i)$	poloha robotu na ose x v daném bodě Globálního s. s.
$x_G(i - 1)$	poloha robotu v předchozím bodě na ose x Globálního s. s.
\dot{x}	rychlost těžiště robotu v ose x Globálního s. s.
$\dot{x}_G(i - 1)$	rychlost těžiště robotu v ose x v předchozím bodě Globálního s. s.
$\dot{x}_{G,i+1}$	rychlost těžiště robotu v ose x v dalším hlavním bodě Globálního s. s.

$y_G(i)$	poloha těžiště robotu v ose y v daném bodě Globálního s. s.
$y_G(i - 1)$	poloha těžiště robotu v předchozím bodě v ose y Globálního s. s.
\dot{y}	rychlost těžiště robotu v ose y Globálního s. s.
$\dot{y}_G(i - 1)$	rychlost těžiště robotu v ose y v předchozím bodě Globálního s. s.
$\dot{y}_{G,i+1}$	rychlost těžiště robotu v ose y v dalším hlavním bodě Globálního s. s.
$\Delta \dot{x}$	přírůstek rychlosti těžiště robotu v ose x Globálního s. s.
$\Delta \dot{y}$	přírůstek rychlosti těžiště robotu v ose y Globálního s. s.

1 Úvod

Robotika jako věda o robotech sahá do čtyřicátých let minulého století, kdy toto slovo použil jako první autor mnoha povídek o robotech Isaac Asimov. Robotika se všeobecně zabývá stavbou robotu a jeho pohybem. Právě pohyb robotu v prostoru je hlavní náplní této diplomové práce. Stará se o něj motorický subsystém robotu, jehož součástí je lokomoční ústrojí, čímž se rozumí podvozek, umožňující pohyb. V praxi se setkáváme s celou řadou druhů podvozků, které se liší vhodností a nebo naopak nevhodností pro jednotlivé účely použití. Nejrozšířenější z nich je podvozek kolový, mezi jehož hlavní výhody patří snadný a rychlý pohyb, nevýhodou je naopak omezená možnost využití ve členitých terénech.

Vzhledem k dlouhé době existence podvozků s klasickými koly lze říci, že tato oblast již prošla důkladným zkoumáním a proto jsme se rozhodli v této práci zaměřit svou pozornost na kola všesměrová, která vznikla jako alternativa ke kolům klasickým. Vhodným uspořádáním všesměrových kol lze zkonstruovat všesměrové lokomoční ústrojí, které umožní pohyb podvozku ve dvou osách.

Ústav automatizace a informatiky se roboty se všesměrovými koly zabývá již několik let, ovšem doposud nebyla zcela ověřena praktická funkčnost navrhovaných kinematických řešení. Tato práce se snaží navrhnout a realizovat funkční kinematické vztahy pro všesměrovou platformu a vytvořit řízení pro mobilní robot se třemi všesměrovými koly. Navrhovaný systém řízení je možno použít s již realizovaným programem pro návrh trajektorie robotu v prostoru [4].

V závěru proběhne zhodnocení funkčnosti a přesnosti navrženého řešení a praktické ověření použitelnosti na novém prototypu všesměrového podvozku.

2 Teoretická analýza dané problematiky

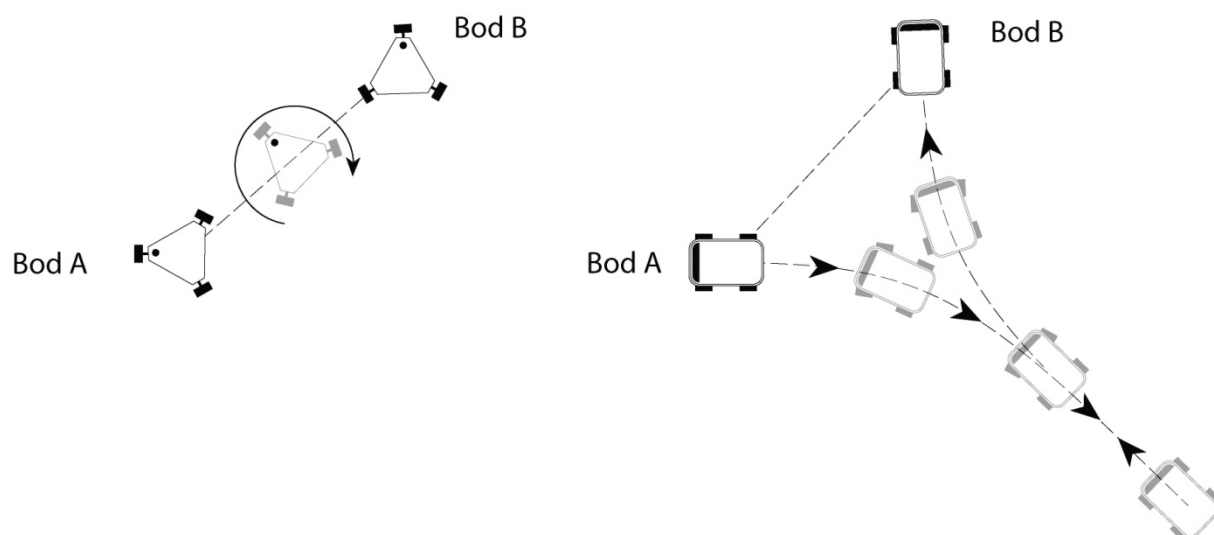
2.1 Všesměrová mobilita

Pro následující kapitolu byly informace čerpány ze zdrojů [5], [6].

Na poli robotiky důležitost mobilních robotů výrazně roste. Díky volnosti jejich pohybu jsou mobilní roboty více flexibilní a mohou vykonávat více úkolů než jejich pevné protějšky. Současné použití mobilních robotů je široké, zahrnuje domácí využití, transport materiálu ve skladištích, nemocnicích, továrnách, ale i průzkum nebezpečných oblastí jako například oceánů, vesmíru či minových polí.

Speciální třídou mobilních robotů jsou všesměrové roboty, které se mohou pohybovat po rovném terénu a jsou schopny translačního a rotačního pohybu okolo svého těžiště, mají tedy tři stupně volnosti. Na rozdíl od klasických mobilních robotů jsou roboty se všesměrovým podvozkem schopny ovládat každý stupeň volnosti nezávisle.

Jak již bylo řečeno, mobilní roboty jsou často navrhovány pro pohyb v rovném terénu, tedy ve 2D. Všesměrový robot má ovšem tři stupně volnosti (x , y , θ) viz Obr. 12. Může se pohybovat podél osy x , y a otáčet se okolo svého těžiště pod úhlem θ . Většina mobilních robotů není schopna řídit tyto stupně volnosti nezávisle na sobě kvůli tzv. neholonomním vazbám. Jako příklad si můžeme uvést dva roboty, které se snaží dostat z bodu A do bodu B viz Obr. 1. Robot bez neholonomních vazeb se může pohybovat všemi směry při rotaci. Takovému pohybu se říká všesměrová mobilita. Oproti tomu robot s klasickým podvozkem potřebuje k dosažení bodu B naplánovat dráhu, která se bude skládat z několika pohybů vpřed a zpět, s cílem dosáhnout požadovaného místa pod určitým úhlem.



Obr. 1 Robot s všesměrovým podvozkem a robot s neholonomními vazbami.

Všesměrová mobilita může být značnou výhodou v případech, kdy je robot nasazen do stísněného nebo přeplněného prostředí.

2.2 Všesměrová kola

Pod pojmem všesměrové kolo rozumíme kolo, které je na obvodu vybaveno valivými elementy ve tvaru válečku nebo soudečku. Elementy se mohou volně otáčet, poháněn je pouze náboj kola [1].

Všesměrové kolo se může pohybovat dvěma směry:

- Rotací kola okolo osy náboje – V tomto případě je pohyb kolmý na osu kola a kolo se pohybuje po obvodové křivce.
- Translací ve směru osy náboje – V tomto případě se otáčí element, který je v kontaktu s podložkou.

Tyto dva pohyby můžeme složit, čímž vznikne pohyb v libovolném směru.

Všesměrová kola můžeme rozdělit do tří základních skupin:

- Univerzální kolo
- Stanfordské kolo
- Kolo Illanator

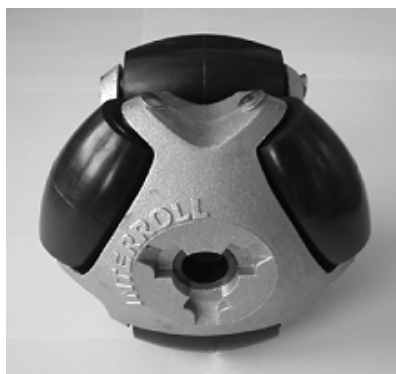
U univerzálního kola mají valivé elementy většinou tvar válečku, u Stanfordského kola tvar soudečku a u kola Illanator se valivé elementy podobají soudečku a jejich osa svírá s osou náboje úhel 45° .

U každého typu lze nalézt výhody i nevýhody. Pojítkem všech je společný cíl, který představuje přiblížení obvodové křivky dokonalé kružnici, čímž se zamezí chvění, vibracím a dalším nežádoucím efektům.

2.2.1 Použitý typ kol

V našem případě byl použit typ Stanfordského kola od firmy Interroll viz Obr. 2. Přejít mezi dvěma sousedními valivými elementy tvoří na vnějším obvodu kola přímku. Tento nedostatek je odstraněn dvěma řadami valivých elementů. V každé řadě jsou tři pasivní válečky [2].

Hlavní geometrický údaj kola, potřebný pro simulaci pohybu, je průměr kola, který činí 8 cm. Průměr valivých elementů není nijak podstatný pro výpočet dráhy, ale může například ovlivňovat schopnost podvozku překonávat překážky.

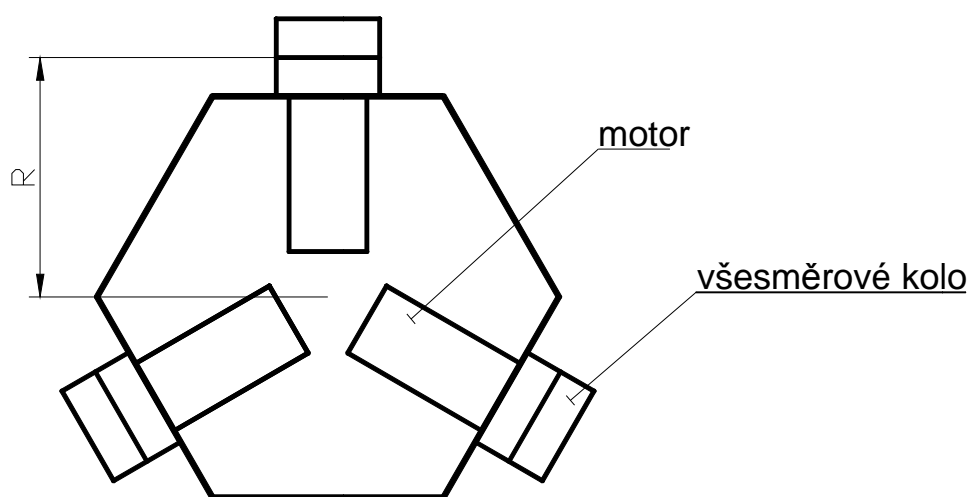


Obr. 2 Použité všesměrové kolo od firmy Interroll.

2.3 Uspořádání podvozku

V našem případě byl uvažován podvozek mobilního robotu ve tvaru pravidelného šestiúhelníku. Ve středu každé z jeho tří stran jsou umístěny motory, které mezi sebou svírají úhel 120° . Jedná se tedy o podvozek se třemi všesměrovými koly, spojnice jejich středů tvoří rovnostranný trojúhelník Obr. 3.

Nejdůležitější geometrický údaj o podvozku je vzdálenost středu kola od těžiště robotu (R). Tento údaj je stejný pro všechna kola. S cílem vytvoření univerzálního programu a jeho použití pro jiný podvozek lze v praktické části tuto hodnotu měnit v simulačním programu. Za základní hodnotu je považováno 183,5mm.



Obr. 3 Schéma podvozku mobilního robotu.

2.4 Použité motory

Poznatky pro tuto kapitolu byly čerpány ze zdrojů [9], [8] a [3].

Pohonem pro jednotlivá kola robotu byly zvoleny motory od firmy Maxon, zabudované v tzv. inteligentním řešení MCD EPOS viz Obr. 4. Zabudovaná jednotka EPOS je založena na standardu CANopen.

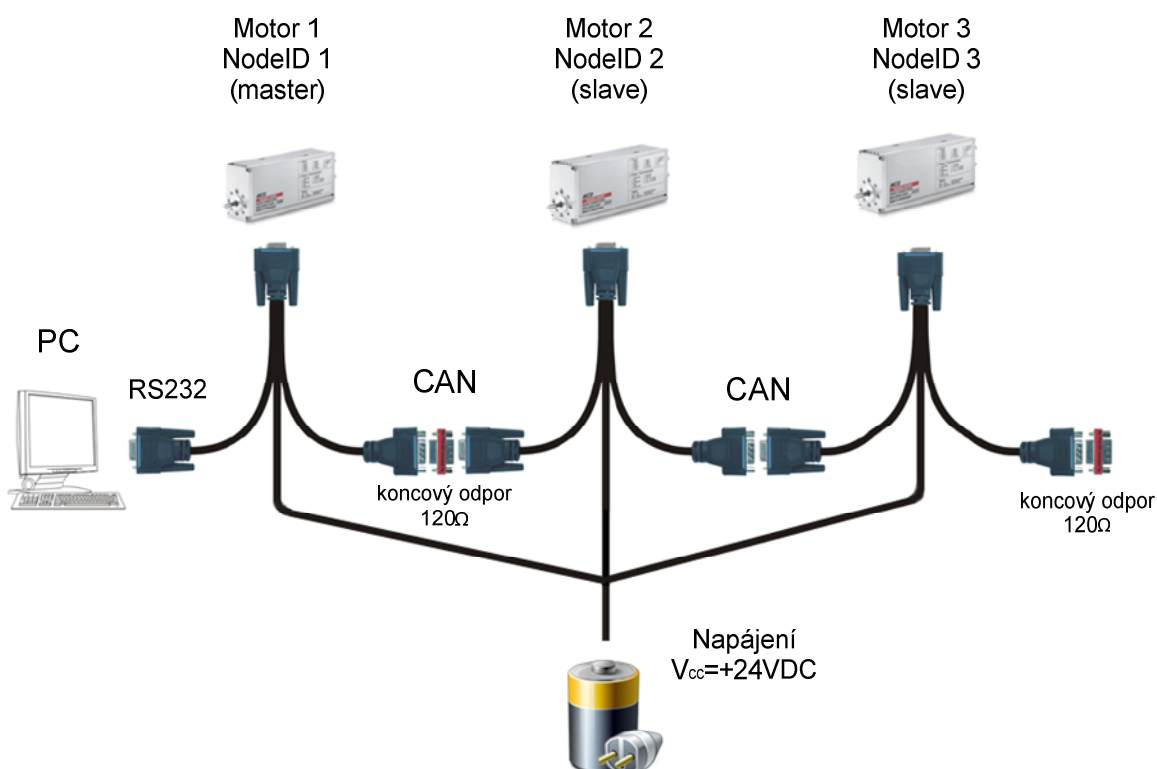
Celý tento inteligentní pohon obsahuje:

- bezkartáčový stejnosměrný elektronicky komutovaný motor o výkonu 60W
- magnetický inkrementální snímač (enkodér)
- řídící jednotku EPOS



Obr. 4 Bezkartáčový stejnosměrný elektronicky komutovaný motor od firmy Maxon [3].

Všechny motory jsou připojeny na sběrnici CAN a jeden z nich slouží jako komunikační brána s PC. Komunikace PC s řídící jednotkou EPOS, která ovládá jednotlivé motory, může probíhat po sériové lince (RS232) nebo po sběrnici CAN přes aplikační protokol CANopen. V našem případě byla použita sériová linka. To znamená, že za použití RS232 lze přistupovat ke všem jednotkám připojeným na sběrnici CAN. Schéma řízení a zapojení jednotek je znázorněno na Obr. 5. Jednotka komunikující s PC přes RS232 je přepnuta do režimu master a ostatní jednotky na sběrnici CAN do režimu slave. Jestliže jednotka EPOS master obdrží od PC příkaz jí určený, vykoná jej. Příkazy adresované jednotkám typu slave jsou jim jednotkou master přeposlány po sběrnici CAN. Každá z jednotek musí mít jedinečné NodeID v rozsahu 1 až 127. Nastavení NodeID, stejně jako změna rychlosti komunikace je provedeno pomocí LSS (Layer Setting Services). V našem případě byly NodeID nastaveny na 1, 2, 3 pro jednotlivé motory a komunikace na maximální hodnoty, tedy pro RS232 na 115.2 kBit/s a pro CAN 1 MBit/s .



Obr. 5 Schéma zapojení a komunikace řídicí stanice s jednotkami.

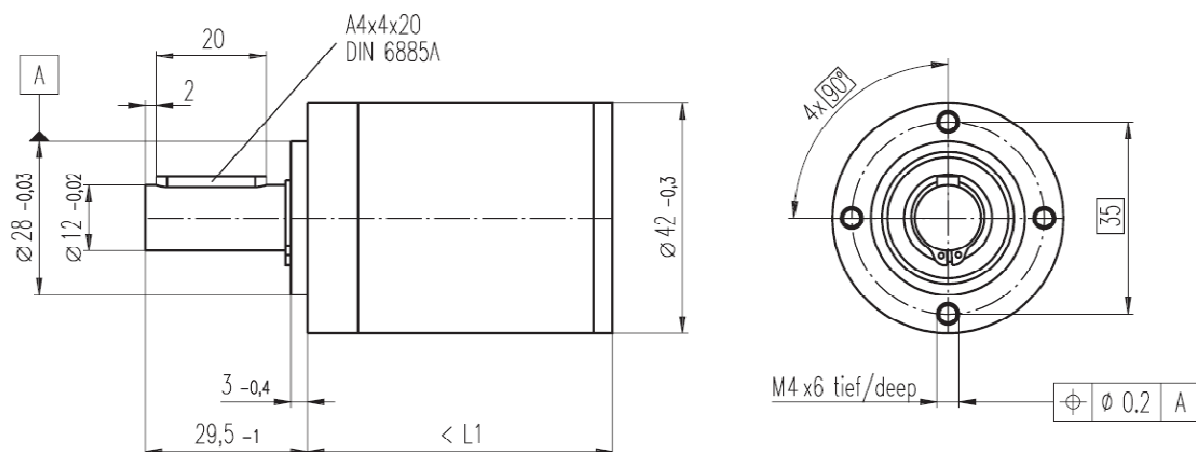
Použité inteligentní pohony mají řadu výhod od jejich snadného použití a řízení přes lineární náběhy rychlostí a drah až k velké spolehlivosti.

2.5 Použitá převodovka

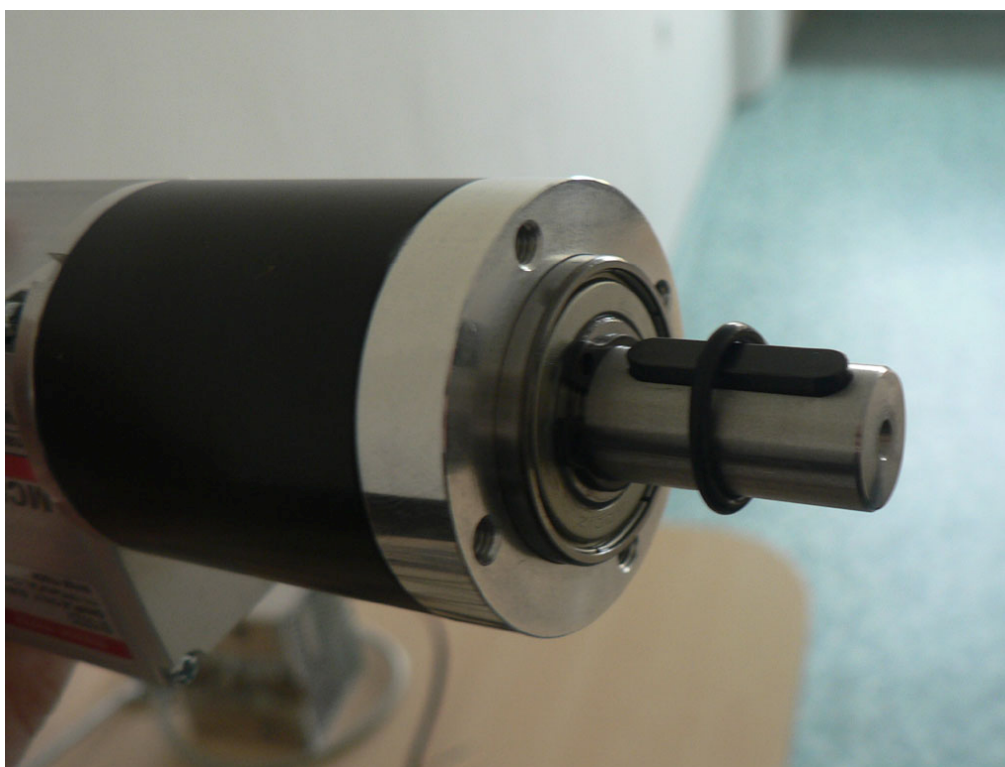
K motoru byla připojena planetová převodovka GP 42 C s převodovým poměrem 15,167 : 1 (převod do pomala), díky které výstupní hřídel získá požadovaný moment při daných otáčkách viz Obr. 8. Rozměry převodovky viz Obr. 7 byly důležité při návrhu nového prototypu podvozku. Převodovka se vyznačuje velmi dobrými parametry, jako je doba životnosti, která dosahuje až několika desítek tisíc hodin. Další předností je maximální doporučená vstupní rychlost, která činí 8000ot/min, což je optimální pro naši aplikaci. Tento typ převodovky je určen pro maximální nepřetržitý moment 15 Nm, což je v porovnání s jejími rozměry nadprůměrná hodnota.



Obr. 6 Ilustrační obrázek planetové převodovky [13].



Obr. 7 Výkres použité planetové převodovky [3].



Obr. 8 Planetová převodovka, umístěná na výstupní hřídeli z jednotky MCD EPOS.

Motor bez převodovky je schopen vyvinout dle specifikace úhlovou rychlost přibližně 8040 ot/min (t. j. 140ot/s) při napájení 24VDC. Pro maximální úhlovou rychlost na výstupní hřídeli z převodovky platí vztah:

$$\omega_{max} = \frac{8040}{15,167 \cdot 60} \cong 9 \text{ ot/s} \quad (2.1)$$

Prakticky budeme využívat maximálně poloviční hodnotu úhlové rychlosti, jelikož zastavit lokomoční ústrojí z takto vysoké hodnoty může trvat i několik sekund s ohledem na zabránění prokluzu kol při zpomalování.

2.6 Parametry řídicí jednotky EPOS a motoru

Jak již bylo zmíněno v předchozí kapitole, inteligentní pohon obsahuje řídicí jednotku EPOS (dále jen jednotka EPOS), která slouží k řízení elektronicky komutovaných motorů vybavených Hallovými sondami a inkrementálním snímačem se sinusovým průběhem proudu. Jednotka je určena pro motory o výkonu 5 až 120W. Elektrická a mechanická data jednotky EPOS a údaje o teplotě na vlhkosti jsou k dispozici v níže uvedených tabulkách.

Tabulka 1 Elektrická data jednotky MCD EPOS [3].

ELEKTRICKÁ DATA	
Napájecí napětí Vcc (zvlnění < 10 %)	+12...+50 V
Max. výstupní napětí	0,9 x Vcc
Max. výstupní proud I _{max} (< 1 s)	9 A
Výstupní proud trvale I _{cont}	2.6 A (T _U =25°C, 5000 rpm)
Vzorkovací frekvence proudového PI regulátoru	10 kHz
Vzorkovací frekvence rychlostního PI regulátoru	1 kHz
Vzorkovací frekvence polohového PID regulátoru	1 kHz
Max. rychlost (2-pólové motory)	12000 min ⁻¹
Zabudovaná tlumivka	15 μH / 5 A
VSTUPY	
Analogové vstupy	2 analog. vstupy 10-bit rozlišení, 0 ... +5 V
Digitální vstupy	4 dig. vstupů
Signály inkrementálního snímače	A, A', B, B', I, I' (max. 1 MHz)
CAN-ID (identifikace CAN uzlu)	Konfigurovatelné DIP přepínačem 1 ... 7
VÝSTUPY	
Digitální výstupy	2 dig. výstupy
NAPĚŤOVÉ VÝSTUPY	
Pomocné napětí	Vcc, max. 1300 mA
Napájení inkrementálního snímače	+5 VDC, max. 100 mA
Napájení Hallových sond	+5VDC, max. 30 mA
INTERFACE	
RS232	RxD; TxD (max. 115 200 bit/s)
CAN	high; low (max. 1 Mbit/s)
INDIKACE	
LED zelená = Enable Stav, červená = Chyba blikající zelená = operační stav	Dvoubarevná LED

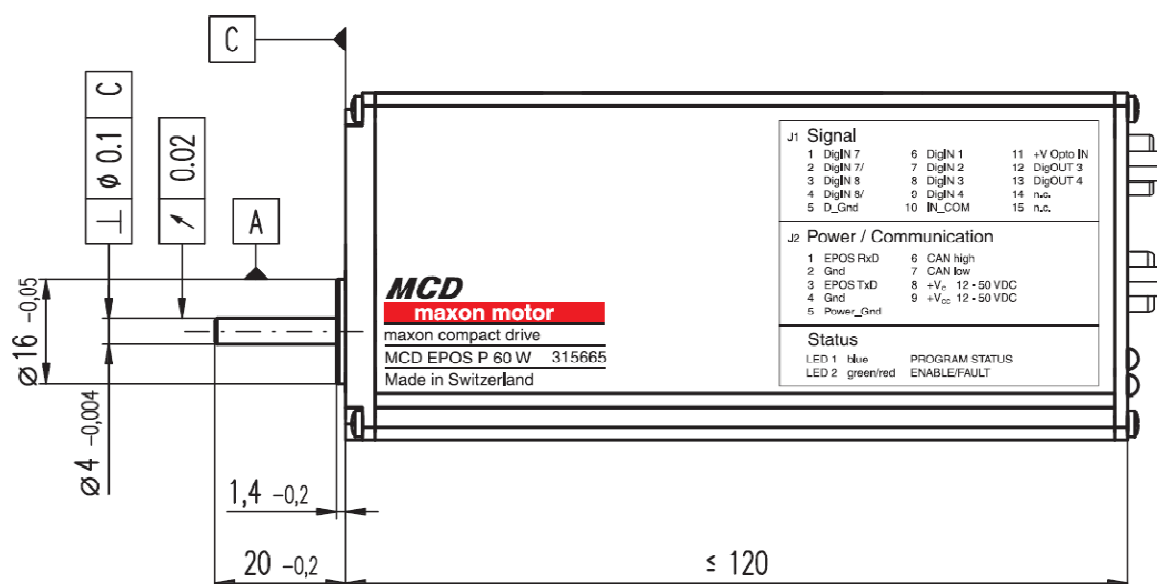
Tabulka 2 Mechanická data jednotky MCD EPOS [3].

MECHANICKÁ DATA	
Hmotnost	495 g
Rozměry	120 x 33 x 53 mm
Upevňovací příruba	pro šrouby M3x4.5

Tabulka 3 Přípustná okolní teplota a vlhkost jednotky MCD EPOS [3].

OKOLNÍ TEPLOTA / VLHKOST	
Provozní teplota	-20 ... +85°C (při teplotě okolí $T_u=25^\circ\text{C}$)
Skladovací teplota	-40 ... +85°C
Bez kondenzace	20 ... 80 %

Boční pohled jednotky MCD EPOS viz Obr. 9.



Obr. 9 Boční pohled jednotky MCD EPOS [3].

2.7 Operační módy jednotky EPOS

Poznátky pro tuto kapitolu byly čerpány ze zdrojů [8] a [10].

Jednotka EPOS má deset módů, z nichž nejdůležitější jsou módy:

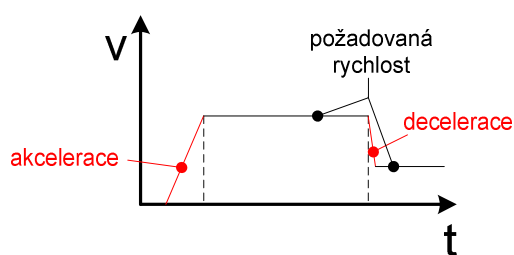
- řízení proudu

- řízení rychlosti
- řízení polohy

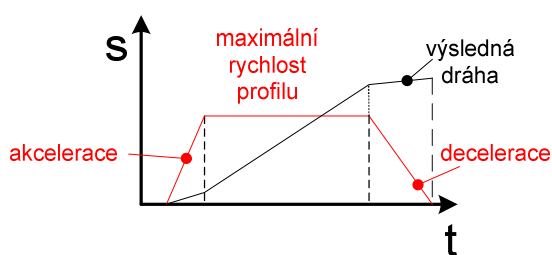
V módu řízení proudu se uplatní pouze regulátor proudu a výkonové zesilovače. Linearita závislosti mechanického momentu na proudu je u motorů od firmy Maxon velmi přesná, takže řízením proudu je řízen moment motoru. Jednotka EPOS přitom omezuje otáčky při odlehčení motoru na vloženou přípustnou mez. [8] Tímto módem se již nebudeme dále zabývat.

V módech řízení rychlosti a polohy si můžeme zvolit druh odezvy na změnu vstupní veličiny. Pokud chceme skokovou změnu rychlosti nebo polohy, zvolíme mód založený na odezvě regulátoru. Pokud chceme požadovanou veličinu změnit s určitým profilem, zvolíme příslušný profilový mód rychlosti nebo polohy. V profilovém módu je zapojen generátor průběhu rychlosti, který vytvoří příslušný profil tak, aby požadovaná hodnota byla dosažena plynulým přechodem.

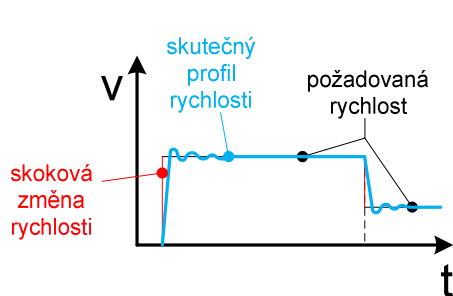
Operační módy



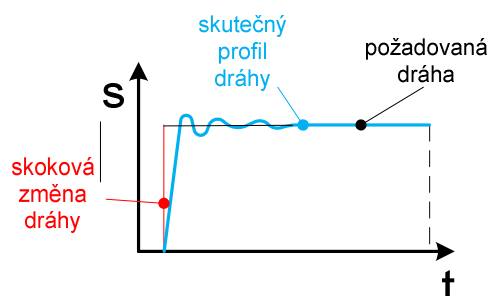
Mód profilu rychlosti



Mód profilu polohy



Mód rychlosti
(odezva regulátoru)



Mód polohy
(odezva regulátoru)

Obr. 10 Čtyři nejdůležitější operační módy jednotky EPOS.

Na Obr. 10 jsou v levém sloupci ukázány dvě reakce na změnu rychlosti. První je s profilem rychlosti a druhá se skokovou změnou. U profilu rychlosti zadáváme před nastavením požadované rychlosti dva parametry a to s jakým zrychlením může rychlost stoupat a s jakým zpomalením klesat. V druhém případě je rychlost měněna dle parametrů regulátoru.

Pravý sloupec ukazuje jiný přístup. Zajímá-li nás absolvování určité dráhy, která je vyjádřena otočením hřídele o určitý počet inkrementů a chceme-li, aby v koncovém bodě dráhy byla rychlost nulová, zvolíme jeden z módů polohy. V módu polohy řízeném odezvou regulátoru je profil rychlosti k dosažení koncového bodu dráhy závislý na parametrech regulátoru. U profilu polohy před zadáním požadované dráhy (ve formě inkrementů) musíme definovat profil rychlosti, který bude respektován během dosažení koncové polohy.

Každý z představených módů má své výhody a nevýhody. Pro případ této diplomové práce nejsou vhodné módy založené na odezvě regulátoru a ani mód profilu polohy. Důvody jsou zřejmé. Jelikož potřebujeme co nejplynulejší přechody mezi jednotlivými úseky dráhy a zřetězený pohyb, který profil polohy neumožňuje, protože rychlost v koncovém bodě je vždy nulová, je jednoznačně profil rychlosti nejlepší volbou.

Budeme-li chtít urazit určitý počet inkrementů v profilu rychlosti za určitý čas, budeme muset nejdříve vypočítat hodnoty akcelerace a decelerace v rovnoměrně zrychleném či v rovnoměrně zpomaleném pohybu. Přesnost polohování v tomto módu závisí na přesnosti časovače v řídicí stanici (PC) a na rychlosti komunikace sériového portu a sběrnice CAN. Čas přístupu k sériovému portu v prostředí operačního systému není zcela zanedbatelný a může významně ovlivnit přesnost polohování. Závěr této práce by měl přinést odpověď i na otázku do jaké míry byla přesnost polohování ovlivněna sériovou komunikací a řízením z prostředí operačního systému.

2.8 Postup při komunikaci s jednotkou EPOS

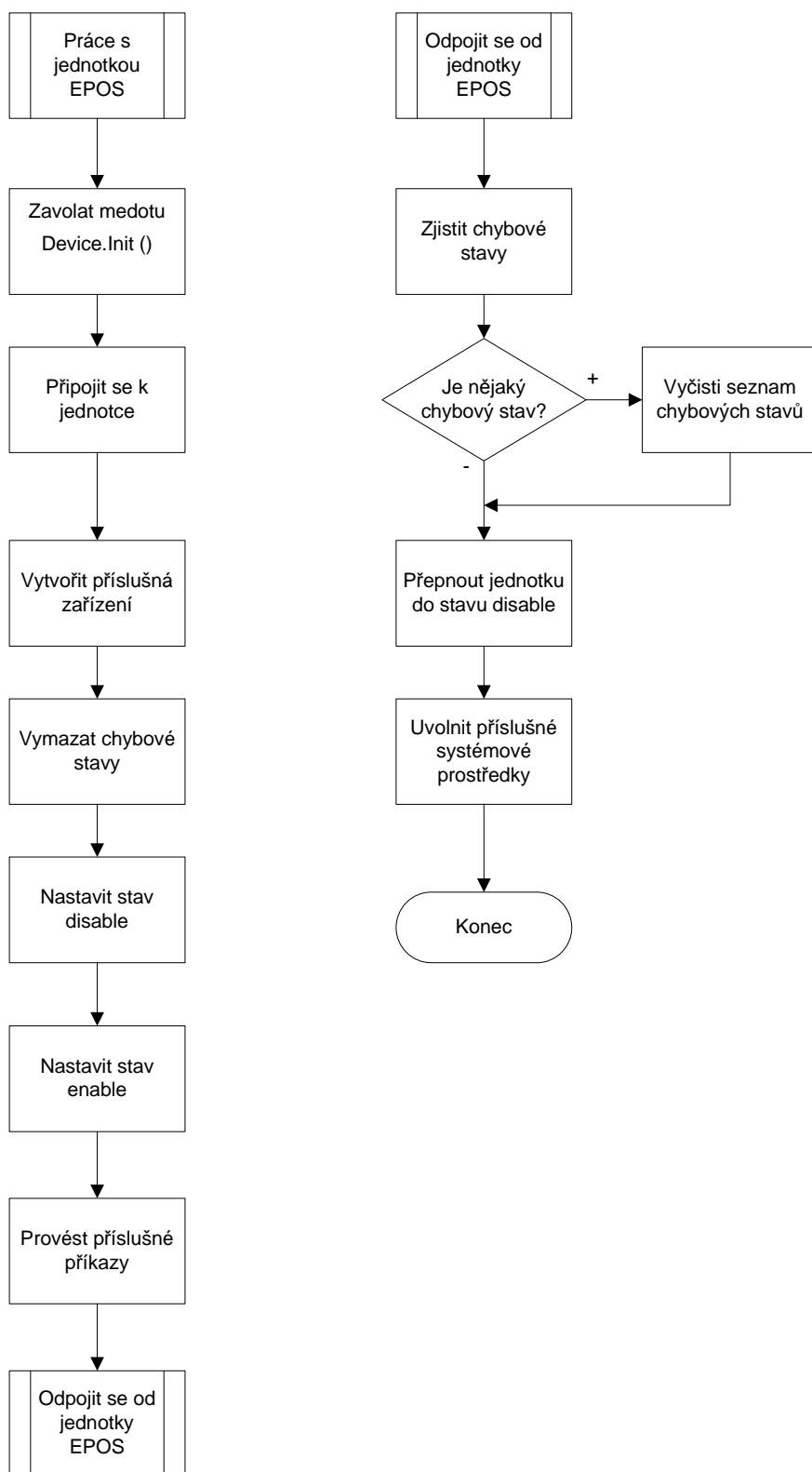
Existuje několik důležitých zásad, které je třeba dodržovat, při práci s jednotkou EPOS. Mezi nejdůležitější patří uvolnění systémových prostředků pomocí volání metod:

- `Device.Cleanup`
- `EposCmd.Net.DeviceManager.Dispose`

Na začátku práce s jednotkou je třeba provést následující kroky v uvedeném pořadí:

- zavolat metodu `EposCmd.Net.VcsWrapper.Device.Init ()`
- vytvořit instanci třídy správce zařízení `EposCmd.Net.DeviceManager`
- vytvořit příslušná zařízení
- vyčistit chybové stavy v každém zařízení
- nastavit všechna zařízení do stavu `disable`
- nastavit všechna zařízení do stavu `enable`

Celý postup při připojení a odpojení je uveden ve vývojovém diagramu viz Obr. 11.



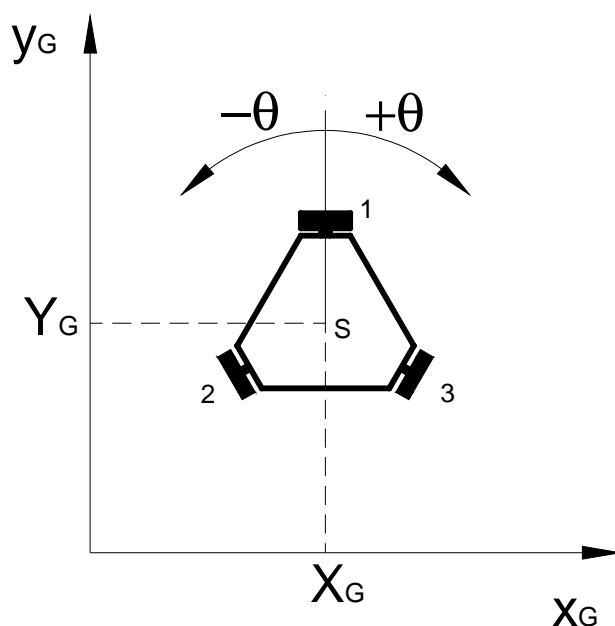
Obr. 11 Vývojový diagram postupu při připojení a odpojení od jednotky EPOS.

3 Praktická část

Praktická část této diplomové práce se skládá z několika oddílů. Na počátku bylo třeba určit souřadné systémy robotu v prostoru a navrhnout kinematické rovnice pro pohyb všesměrového podvozku robotu. Dále byl tento model simulován v prostředí aplikace Robot path. Po úspěšné simulaci byla navržena knihovna sériové komunikace mezi řídicí stanicí (PC) a robotem. Poté následovalo otestování teoretického návrhu a vyvození závěrů. Přesnost požadované dráhy robotu byla ověřena údaji z enkodérů motorů a inerciálními snímači.

3.1 Zavedení souřadných systémů

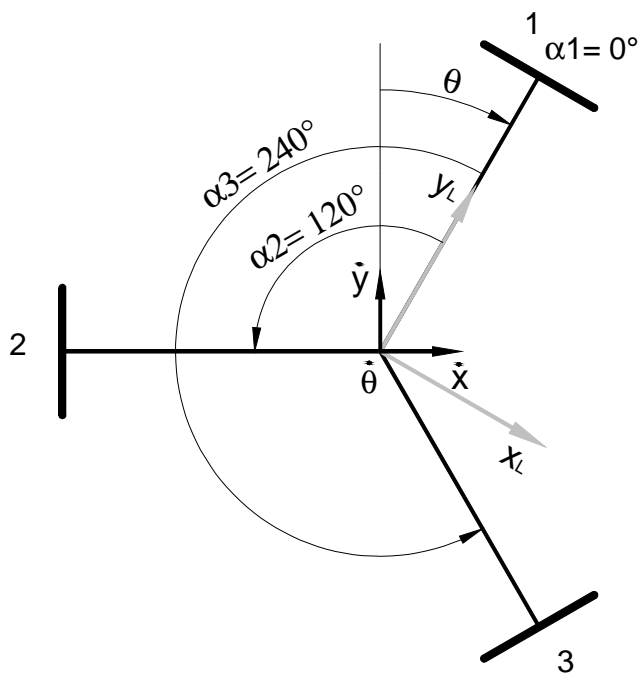
Nejdůležitějším souřadným systémem pro pohyb robotu je Globální souřadný systém Obr. 12, který reprezentuje prostředí robotu a udává jeho jednoznačnou polohu v prostředí. Skládá se ze souřadnic x , y a úhlu θ . Úhel θ reprezentuje natočení robotu vůči ose y . Představme si úsečku vedenou z těžiště robotu (S) do středu kola jedna. Pokud je tato úsečka rovnoběžná s osou y , je úhel θ nulový. Kladný úhel je pak dán natočením robotu po směru hodinových ručiček. Takto zavedený souřadný systém je shodný s určením polohy robotu v předešlé diplomové práci [4]. Bude tedy možno v simulačním programu provést načtení dráhy z programu Mobile robot studio, který slouží k navrhování trajektorie robotu.



Obr. 12 Globální souřadný systém a poloha robotu v něm.

Dalším souřadným systémem je Lokální souřadný systém (x_L, y_L) , který se vztahuje k robotu samotnému. Jeho počátek se shoduje s těžištěm robotu (S), přičemž osa y_L je totožná se spojnici těžiště robotu a středem kola 1. V tomto souřadném systému jsou umístěna tři všesměrová kola pod úhly α_1 , α_2 , α_3 . Počátkem úhlů je kladná osa y_L a jejich směr je proti

směru hodinových ručiček (opačně než úhel θ). Jednotlivé úhly mají velikosti $\alpha_1=0^\circ$, $\alpha_2=120^\circ$ a $\alpha_3=240^\circ$.

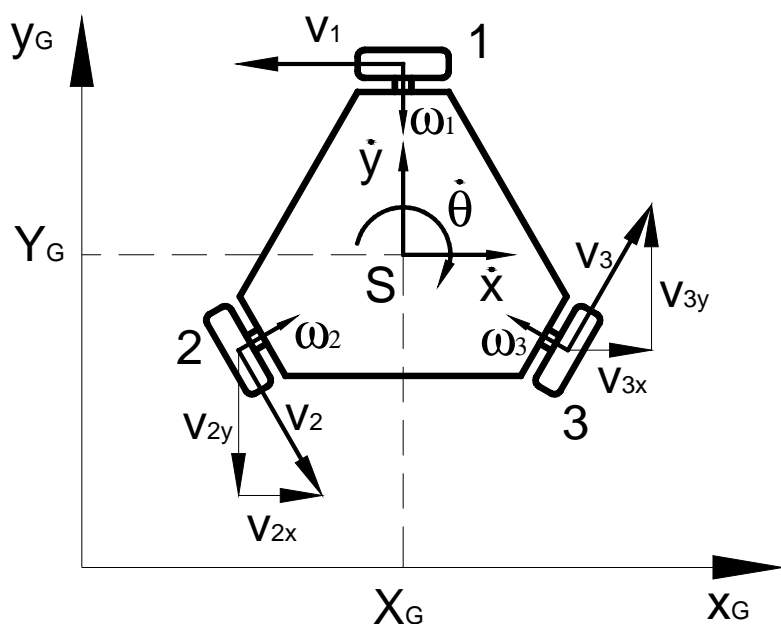


Obr. 13 Lokální souřadný systém a umístění kol v něm.

3.2 Vektory rychlostí kol

Před řešením inverzní úlohy kinematiky robotu je třeba stanovit směry rychlostí jednotlivých kol a jejich rozložení do os Globálního souřadného systému. V našem případě představují obvodové rychlosti jednotlivých kol (v_1, v_2, v_3) a úhlové rychlosti ($\omega_1, \omega_2, \omega_3$) hlavní neznámé při řešení inverzní úlohy kinematiky. Směry vektorů úhlových rychlostí jsou určeny podle pravidla pravé ruky a můžeme tedy podle nich identifikovat smysl otáčení kol. Na Obr. 14 jsou znázorněny všechny důležité vektory rychlostí tří kol robotu.

Každý vektor rychlosti kola lze rozložit do os x_G a y_G . V dolním indexu vektoru rychlosti je uvedeno číslo kola a případně i souřadnice, do které je vektor promítnut. Záporné znaménko ve výsledku tedy značí opačnou orientaci vektoru rychlosti. Rychlost celé základny robotu budeme značit dle Newtonovy notace $(\dot{x}, \dot{y}, \dot{\theta})$. Hodnoty představují rychlosti v ose x_G a y_G . $\dot{\theta}$ označuje úhlovou rychlost s jakou se otáčí těžiště robotu, kladný směr je znázorněn na Obr. 14.



Obr. 14 Stanovení kladných směrů rychlostí a jejich rozložení do os.

3.3 Inverzní úloha kinematiky

Kinematické úvahy vycházející z literatury [5] a [6] byly rozšířeny a aplikovány na daný model.

Vztahy mezi rychlostí celé platformy $(\dot{x}, \dot{y}, \dot{\theta})$ a rychlostí středů jednotlivých kol (v_1, v_2, v_3) řeší inverzní úloha kinematiky.

Rychlost všesměrového kola v_j můžeme rozdělit na část čistě translačního a čistě rotačního pohybu robotu viz rovnice (3.1).

$$v_j = v_{tran,j} + v_{rot} \quad (3.1)$$

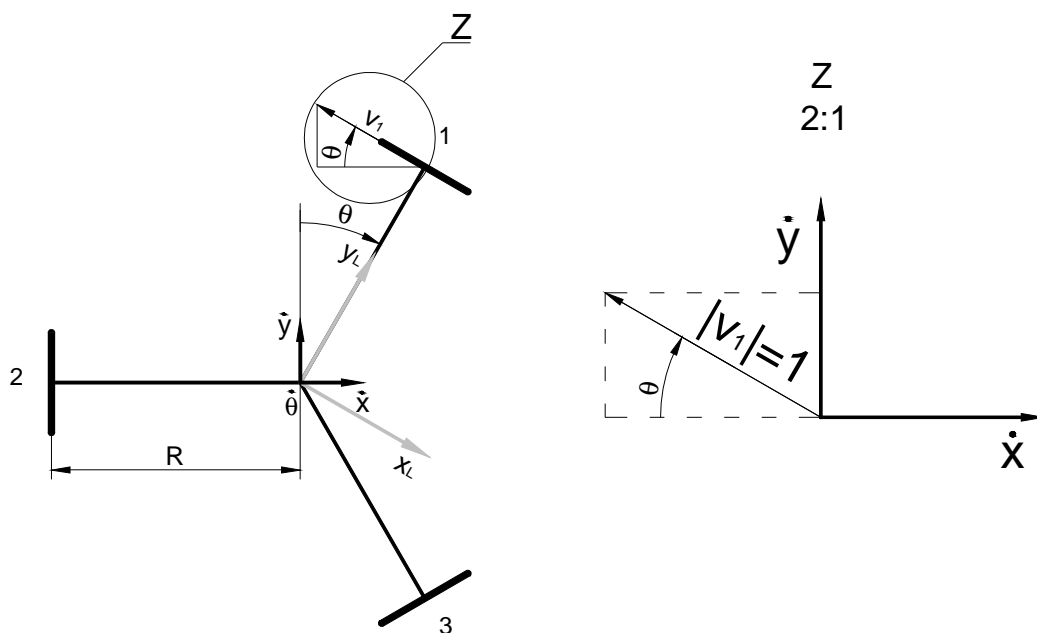
Nejprve se zabývejme jen translačním pohybem.

Mějme jednotkový vektor vektoru rychlosti kola jedna (v_1) viz Obr. 15. Tento vektor promítneme do vektorů \dot{x}, \dot{y} a obdržíme rovnici (3.4).

$$\cos(\theta) = \frac{-\dot{x}}{1} \rightarrow \dot{x}_1 = -\cos(\theta) \quad (3.2)$$

$$\sin(\theta) = \frac{\dot{y}}{1} \rightarrow \dot{y}_1 = \sin(\theta) \quad (3.3)$$

$$v_{1,tran} = -\cos(\theta) \dot{x} + \sin(\theta) \dot{y} \quad (3.4)$$



Obr. 15 Znáznornění rychlosti kola 1 a promítnutí jednotkového vektoru.

Daný postup můžeme aplikovat i na další kola, pokud vezmeme v úvahu, že jsou umístěna vůči prvnímu s offsetem α_2 a α_3 . Ten má ovšem opačný směr než úhel θ , a proto je třeba jej od úhlu θ odečíst.

$$v_1 = -\cos(\theta - \alpha_1) \dot{x} + \sin(\theta - \alpha_1) \dot{y} \quad (3.5)$$

$$v_2 = -\cos(\theta - \alpha_2) \dot{x} + \sin(\theta - \alpha_2) \dot{y} \quad (3.6)$$

$$v_3 = -\cos(\theta - \alpha_3) \dot{x} + \sin(\theta - \alpha_3) \dot{y} \quad (3.7)$$

Po dosazení konstant mají rovnice následující tvar:

$$v_1 = -\cos(\theta) \dot{x} + \sin(\theta) \dot{y} \quad (3.8)$$

$$v_2 = -\cos\left(\theta - \frac{2}{3}\pi\right) \dot{x} + \sin\left(\theta - \frac{2}{3}\pi\right) \dot{y} \quad (3.9)$$

$$v_3 = -\cos\left(\theta - \frac{4}{3}\pi\right) \dot{x} + \sin\left(\theta - \frac{4}{3}\pi\right) \dot{y} \quad (3.10)$$

Při čisté rotaci robotu musí být splněna podmínka v rovnici (3.11).

$$v_{rot} = R\dot{\theta} \quad (3.11)$$

Po dosazení rovnic (3.8) až (3.11) do rovnice (3.1) dostaneme konečnou podobu kinematických rovnic (3.12) – (3.14).

$$v_1 = -\cos(\theta) \dot{x} + \sin(\theta) \dot{y} + R\dot{\theta} \quad (3.12)$$

$$v_2 = -\cos\left(\theta - \frac{2}{3}\pi\right) \dot{x} + \sin\left(\theta - \frac{2}{3}\pi\right) \dot{y} + R\dot{\theta} \quad (3.13)$$

$$v_3 = -\cos\left(\theta - \frac{4}{3}\pi\right) \dot{x} + \sin\left(\theta - \frac{4}{3}\pi\right) \dot{y} + R\dot{\theta} \quad (3.14)$$

Nyní máme definované vztahy mezi obvodovou rychlostí kol a rychlostí všesměrové platformy v Globálním souřadném systému. Ale můžeme jít ještě dál. Obvodová rychlost kola se vztahuje k úhlové rychlosti kola ω_j dle rovnice,

$$v_j = r\omega_j \quad (3.15)$$

kde r značí poloměr všesměrového kola. Pak má výsledný vztah podobu:

$$\omega_j = \frac{-\cos(\theta - \alpha_j) \dot{x} + \sin(\theta - \alpha_j) \dot{y} + R\dot{\theta}}{r} \quad (3.16)$$

Velikost úhlové rychlosti kola ω nemůže být ovšem neomezená, proto je třeba při výpočtu počítat s omezujícími podmínkami:

$$\omega_{min} \leq \omega_j(i) \leq \omega_{max} \quad \text{pro } j = 1,2,3; i = 0,1,\dots,N-1 \quad (3.17)$$

kde: $i = N \cdot \Delta t$. Hodnota ω_{max} vyplývá z rovnice (2.1) a pro ω_{min} platí:

$$\omega_{min} = -\omega_{max} \quad (3.18)$$

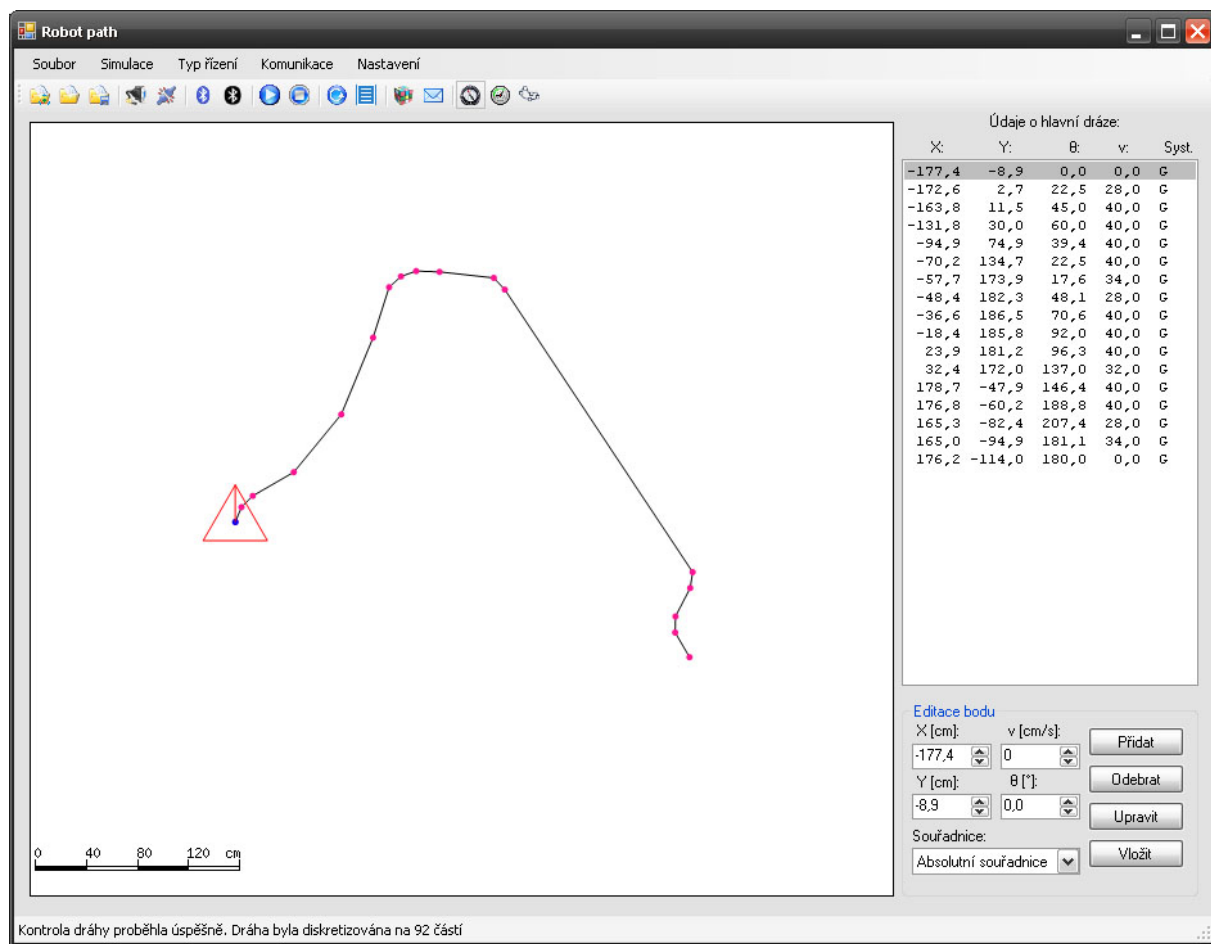
3.4 Aplikace Robot path

Aplikace Robot path viz Obr. 16 slouží jako simulační software pro návrh trajektorie mobilního robotu a můžeme ho pomocí zmínění aplikace i přímo ovládat. Aplikace byla vytvořena ve vývojovém prostředí Microsoft Visual Studio 2008 v programovacím jazyku C# pro platformu .NET 2.0. Verze 2.0 platformy .NET byla zvolena s ohledem na větší rozšířenost než novější verze této platformy.

Během tvorby této aplikace bylo využíváno služeb Subversion hostingu Codesion[11].

Volba verzovacího systému Subversion (SVN) v kombinaci s hostingem měla tyto výhody:

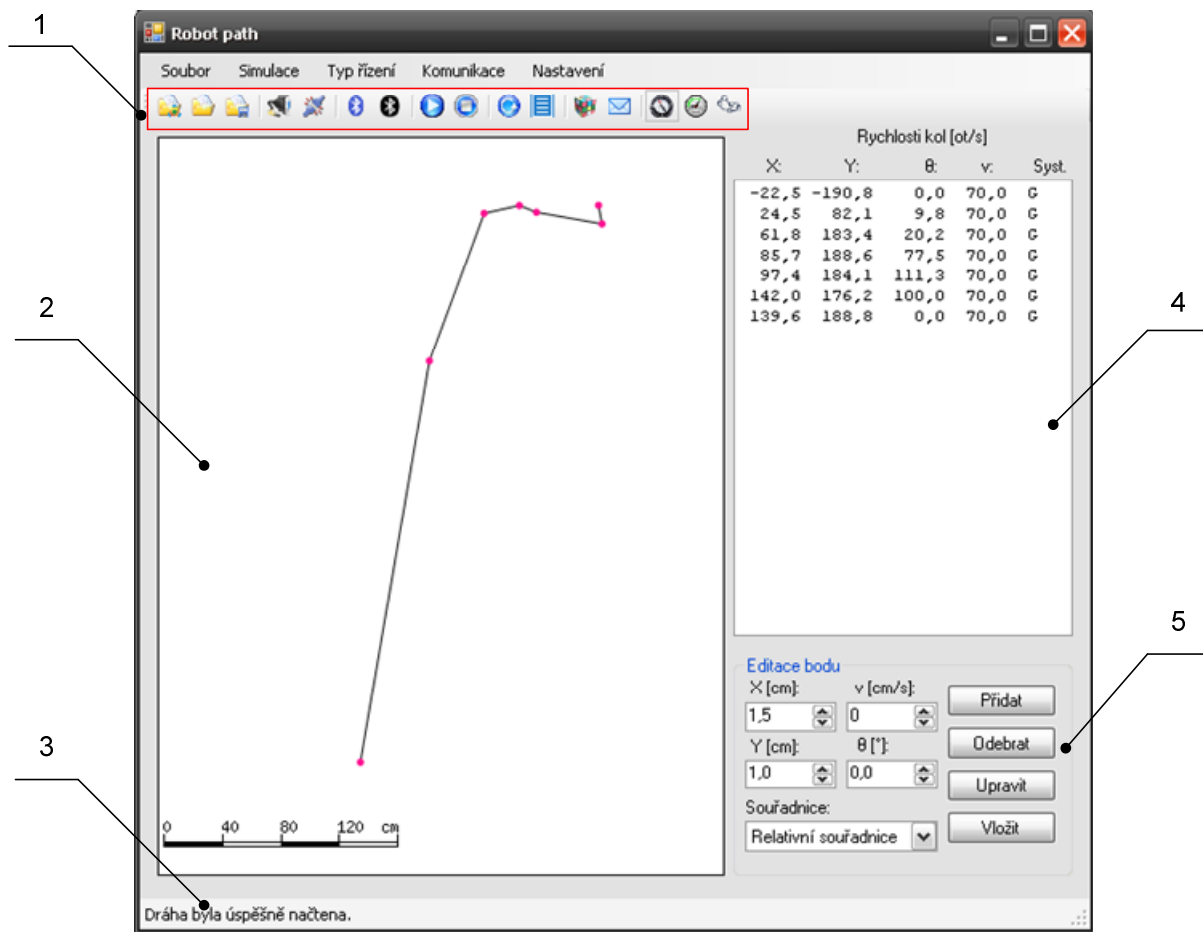
- snadná správa verzí programu
- přehled o provedených změnách každý den v podobě commit komentářů
- veškeré zdrojové kódy dostupné online a na jednom místě
- bezpečnost
- nemožnost odcizení nebo ztráty zdrojových kódů
- snadná spolupráce s vývojovým prostředím Visual Studio (Add-in AnkhSVN)



Obr. 16 Aplikace Robot path.

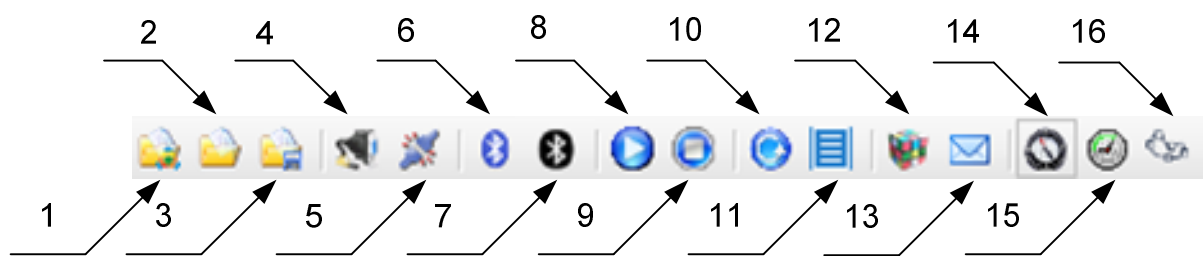
3.4.1 Hlavní okno aplikace

Hlavní okno s popisem je znázorněno na obrázku Obr. 17. V levé části hlavního okna se nachází panel (2) znázorňující dráhu robotu. Růžové body značí hlavní body dráhy, kterou má robot opsat. Celá dráha je v měřítku, kde jeden pixel na obrazovce odpovídá jednomu centimetru ve skutečnosti. Simulace pohybu robotu je řešena ve 2D prostředí. Ve spodní části se nachází stavový řádek (3), určený pro výpis všech důležitých hlášení programu včetně aktuálního stavu nebo případných chyb. Jednotlivé body dráhy můžeme editovat v editačním poli (5), které se nachází přímo pod výpisem (4) všech hlavních bodů dráhy.



Obr. 17 Hlavní okno s popisem.

V horní části hlavního okna se nachází panel nástrojů (1), na němž najdeme nejdůležitější nástroje pro simulaci a řízení robotu. Jednotlivé položky jsou blíže znázorněny na Obr. 18.



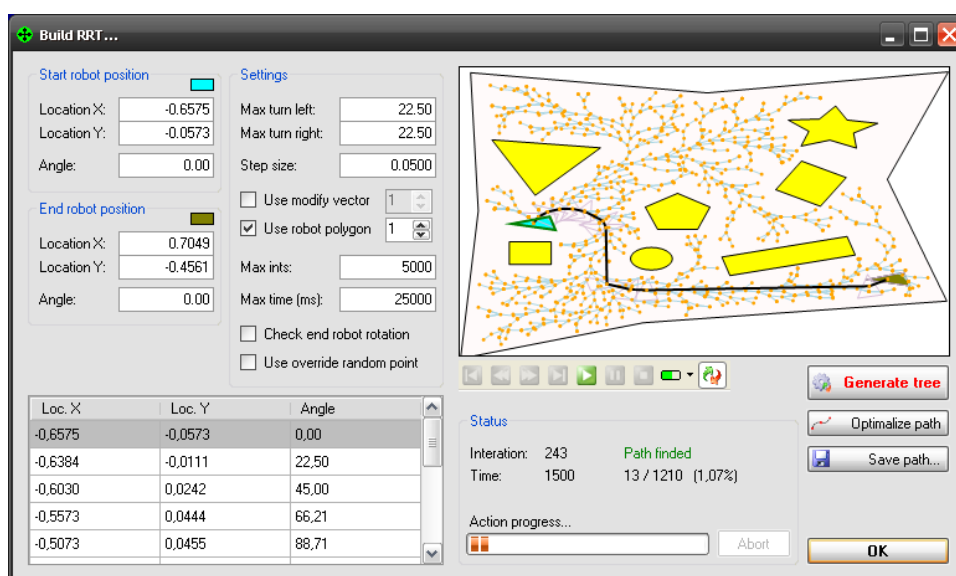
Obr. 18 Popis panelu nástrojů.

- 1 – Načíst dráhu robotu z XML souboru programu Mobile robot studio
- 2 – Načíst dráhu ze souboru Path.xml
- 3 – Uložit dráhu do souboru Path.xml
- 4 – Připojit se k robotu
- 5 – Odpojit se od robotu
- 6 – Připojit se k inerciálním snímačům přes BlueTooth

- 7 – Zrušit spojení přes BlueTooth
- 8 – Spustit simulaci
- 9 – Zastavit simulaci
- 10 – Opakovat dráhu robotu
- 11 – Zobrazit/skrýt diskretizované body
- 12 – Diskretizovat dráhu a řešit inverzní úlohu kinematiky
- 13 – Diskretizovat dráhu, řešit inverzní úlohu kinematiky a zahájit ovládání pohonů
- 14 – Zobrazit ve výpise údaje o hlavních bodech
- 15 – Zobrazit ve výpise údaje o rychlostech všech motorů v jednotlivých úsecích
- 16 – Zobrazit ve výpise údaje o diskretizovaných bodech

3.4.2 Načtení dráhy z Mobile robot studio

Na Ústavu automatizace a informatiky byla vypracována diplomová práce, zabývající se plánováním trajektorie pohybu robotu za použití RRT algoritmu. Výsledkem této práce je program Mobile robot studio, ve kterém je možno naplánovat a optimalizovat trajektorii viz Obr. 19.



Obr. 19 Plánování dráhy v programu Mobile robot studio [4].

Tuto dráhu je možno exportovat do XML souboru, který lze načíst v programu Robot path a danou dráhu vykonat. Ukázka XML souboru dráhy:

```
<?xml version="1.0" encoding="utf-8"?>
<root>
  <project fileName="../../Data/Example.mrp" />
  <path>
    <conf angle="0.0000" x="-0.7095" y="-0.0354" />
    <conf angle="0.3927" x="-0.6904" y="0.0108" />
    <!-- Další body -->
    <conf angle="3.6206" x="0.6611" y="-0.3294" />
    <conf angle="3.1603" x="0.6601" y="-0.3794" />
    <conf angle="0.0000" x="0.7049" y="-0.4561" />
  </path>
</root>
```



```
</path>
</root>
```

Po načtení dráhy je třeba zadat v dialogovém okně použité měřítko pro velikost souřadnic x a y , jelikož jednotky jsou bezrozměrné. Dále je třeba definovat rychlosti v jednotlivých hlavních bodech dráhy, které rovněž nejsou známy.

3.4.3 Uložení dráhy do XML

Jakmile máme navrženou dráhu robotu a rozhodneme se opustit program, budeme vyzváni k jejímu uložení. Možností k uložení je několik. Rozhodneme-li se pro rychlé uložení do souboru Path.xml, bude tato dráha po příštím spuštění aplikace automaticky načtena. Druhou možností je „uložit jako“, kde budeme vyzváni k zadání vlastního názvu souboru. Formát XML souboru obsahuje následující údaje o poloze a rychlosti robotu na dráze:

- Souřadnici x těžiště robotu v příslušném souřadném systému v centimetrech
- Souřadnici y těžiště robotu v příslušném souřadném systému v centimetrech
- Koncová rychlost, kterou má robot mít v daném bodě, zadaná v centimetrech za sekundu
- Úhel θ v radiánech
- Typ souřadného systému, ve kterém jsou souřadnice (x, y, θ) zadány

K uložení a načtení dat byla použita serializace a deserializace objektu, jež nám umožňují snadnou a rychlou práci s daty.

```
<?xml version="1.0" encoding="utf-8"?>
<RobotPath xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Points>
    <point>
      <_X>50</_X>
      <_Y>75</_Y>
      <EndSpeed>0</EndSpeed>
      <_angle>0.78539816339744828</_angle>
      <TypeOfFrame>globalFrame</TypeOfFrame>
    </point>
    <!--          Další body          -->
    <point>
      <_X>15</_X>
      <_Y>30</_Y>
      <EndSpeed>11</EndSpeed>
      <_angle>1.5707963267948966</_angle>
      <TypeOfFrame>localFrame</TypeOfFrame>
    </point>
  </Points>
</RobotPath>
```

3.4.4 Nastavení technických parametrů

Pro správnou činnost aplikace je třeba definovat celou řadu technických údajů, nejčastěji se týkajících samotného robotu, komunikace či typu řízení. Některé z nich bylo třeba uložit, aby byly přístupné i v budoucnu. Mezi nejdůležitější z nich patří:

- Průměr všesměrového kola (cm)

- Počet pulsů na otáčku kola
- Vzdálenost od těžiště robotu ke středu kola - R [cm]
- Maximální úhlová rychlost kola - ω_{\max} [ot/s]
- Výchozí port
- Čas diskretizace - Δt [setiny sekundy]
- Převodový poměr
- Maximální zrychlení kola - a_{\max} [cm/s^2]

Změnu výše uvedených parametrů lze provést v příslušných roletách programu nebo pomocí editace XML souboru Settings.xml, do něhož jsou data ukládána. Jeho obsah je uveden níže.

```
<Settings>
  <WheelDiameter>80</WheelDiameter>
  <PulsesPerRevolution>4000</PulsesPerRevolution>
  <R>18.5</R>
  <ACKTimeout>100</ACKTimeout>
  <BusyTimeout>200</BusyTimeout>
  <MaxAngularVelocityOfWheel>10</MaxAngularVelocityOfWheel>
  <DefaultPort>COM9</DefaultPort>
  <DiscreteTime>50</DiscreteTime>
  <GearRatio>13</GearRatio>
  <PeripheralAcceleration>100</PeripheralAcceleration>
</Settings>
```

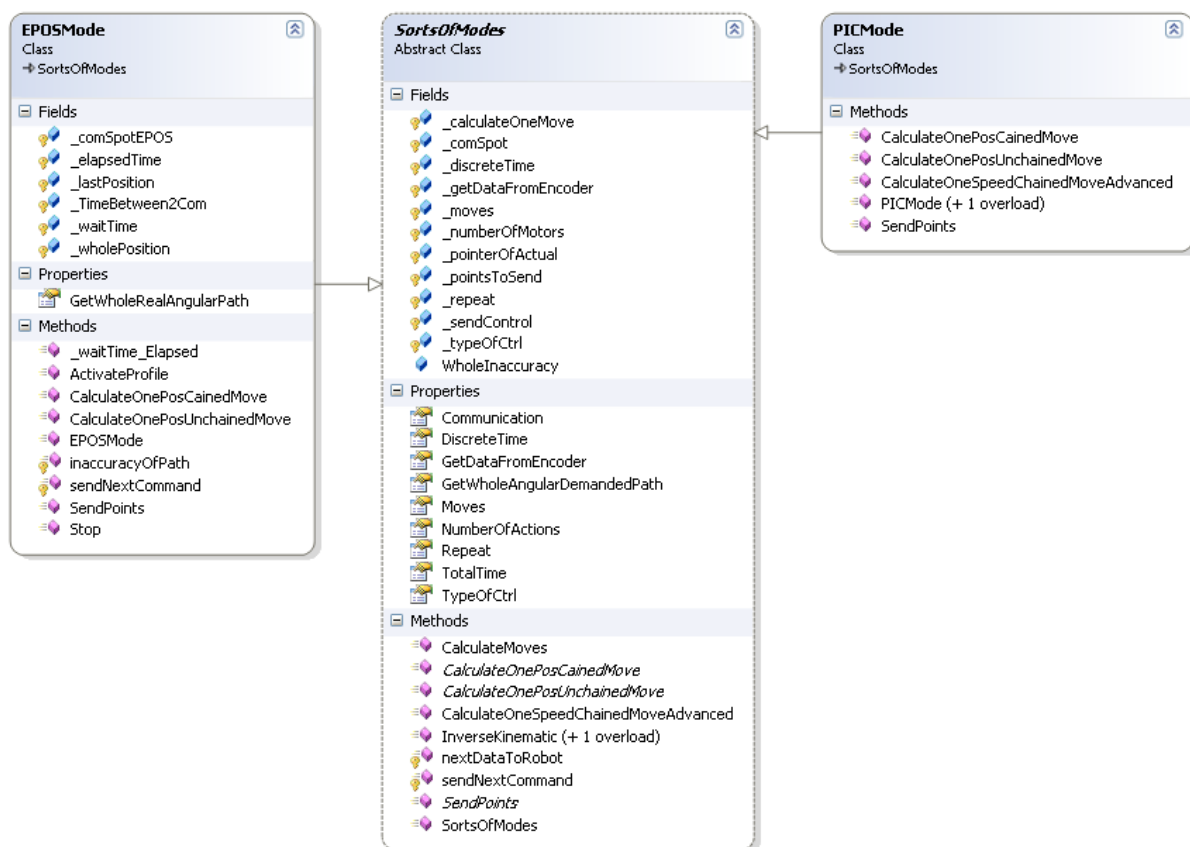
3.4.5 Diagramy nejdůležitějších tříd

Diagram tříd zobrazuje statickou strukturu tříd a vztahů mezi nimi [12].

Aplikace Robot path zahrnuje jeden spustitelný soubor a několik dynamicky linkovaných knihoven. Nejdůležitější knihovnou je CmdFeatures.dll, která obsahuje většinu matematických tříd a struktur používaných v programu. Její součástí je několik tříd určených ke komunikaci s jednotkou EPOS a dalšími zařízeními, jako například akcelerometry či starší řídicí jednotkou robotu. Dále se v této knihovně nachází vlastní třída výjimek, která je s touto knihovnou úzce spjata.

Nejprve bych představil třídy druhů režimů, ve kterých řídicí aplikace může pracovat. Režimem je myšleno, jaké zařízení bude aplikace řídit.

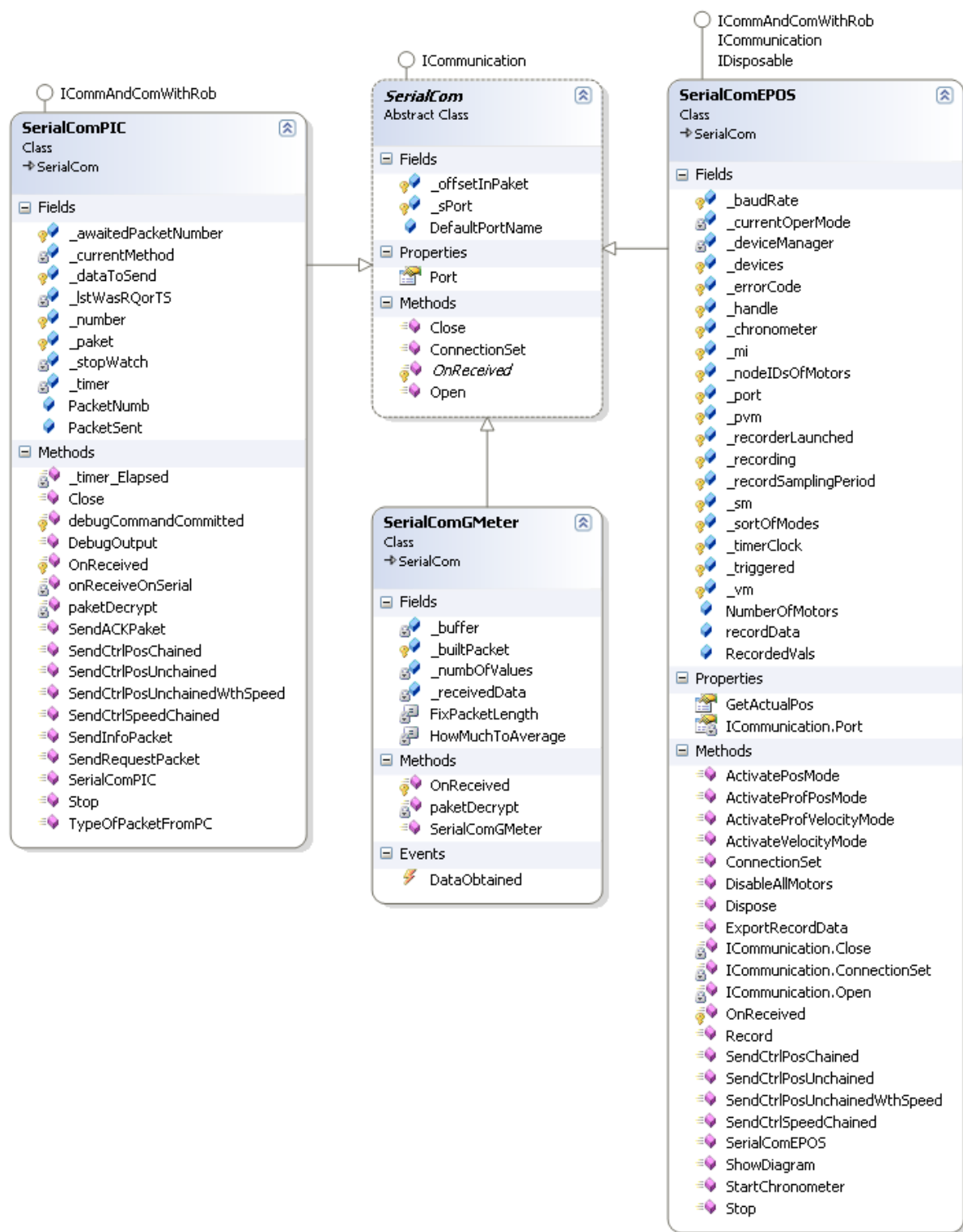
Nejprve se věnujme třídám, které se zabývají řízením. Snahou bylo udržet co největší univerzálnost řízení, a tedy umožnit aplikovat zvolené řešení jak na novou platformu založenou na jednotkách EPOS tak i na starší model založený na PIC procesorech a PWM řízení. Jedná se o tři třídy, z nichž jedna je abstraktní a zbylé dvě jsou od ní odvozeny. Hlavním úkolem těchto tříd je vypočítat z příslušných bodů dráhy údaje o rychlostech a vydávat povely k jejich vyslání. Jedním z druhotných úkolů je mimo jiné i uchovávat informace z enkodérů a počítat přesnost dráhy. Diagram těchto tříd se nachází na Obr. 20.



Obr. 20 Diagram tříd určených k řízení

Druhou neméně důležitou hierarchií tříd jsou třídy sériové komunikace viz Obr. 21. Bázovou třídou je abstraktní třída obecné sériové komunikace, která implementuje rozhraní ICommunication. Další třídy od ní odvozené opět implementují další rozhraní podle typu zařízení, se kterým komunikují. Celkem se jedná o tři druhy zařízení, se kterými lze komunikovat po sériové lince. Prvním z nich představuje jednotku EPOS, druhým je jednotka s akcelerometry a dále se jedná o starší verze řídicí jednotky robotu, založenou na PIC procesorech. Třída SerialComEPOS navíc musí implementovat rozhraní IDisposable, jelikož pracuje s neřízenými prostředky, které bylo nutné po ukončení práce uvolnit.

K veškerým třídám, jejich členům, vlastnostem, metodám, jejich parametrům a návratovým hodnotám byla vytvořena elektronická dokumentace, která je dostupná na příloženém CD.



Obr. 21 Diagram tříd sériové komunikace.

3.5 Diskretizace dráhy

Při tvorbě této kapitoly byl brán v úvahu zdroj [7].

Kinematické vztahy popsané v rovnicích (3.12) až (3.14) počítají s konstantním úhlem θ , který je ovšem konstantní jen při translačním pohybu. Pokud chceme vykonat pohyb robotu

v prostoru zároveň s jeho rotací, je třeba rozdělit celkový časový interval $[0, t_f]$ do N stejných časových úseků, ve kterých můžeme úhel θ považovat za konstantní.

$$t_i - t_{i-1} = \Delta t = \frac{t_f}{N} \quad \text{pro } i = 1, 2, \dots, N \quad (3.19)$$

Jestliže považujeme rychlosti, jak úhlové tak translační, za rovnoměrně zrychlené v každém úseku, platí:

$$x_G(i) = x_G(i-1) + \frac{1}{2} \dot{x}_G(i-1) \cdot \dot{x}_{G,i+1} \cdot \Delta t \quad (3.20)$$

$$y_G(i) = y_G(i-1) + \frac{1}{2} \dot{y}_G(i-1) \cdot \dot{y}_{G,i+1} \cdot \Delta t \quad (3.21)$$

$$\theta_G(i) = \theta_G(i-1) + \left(\theta_{G,i+1} - \theta_G(i-1) \right) \frac{s_i}{s_{i+1}} \quad (3.22)$$

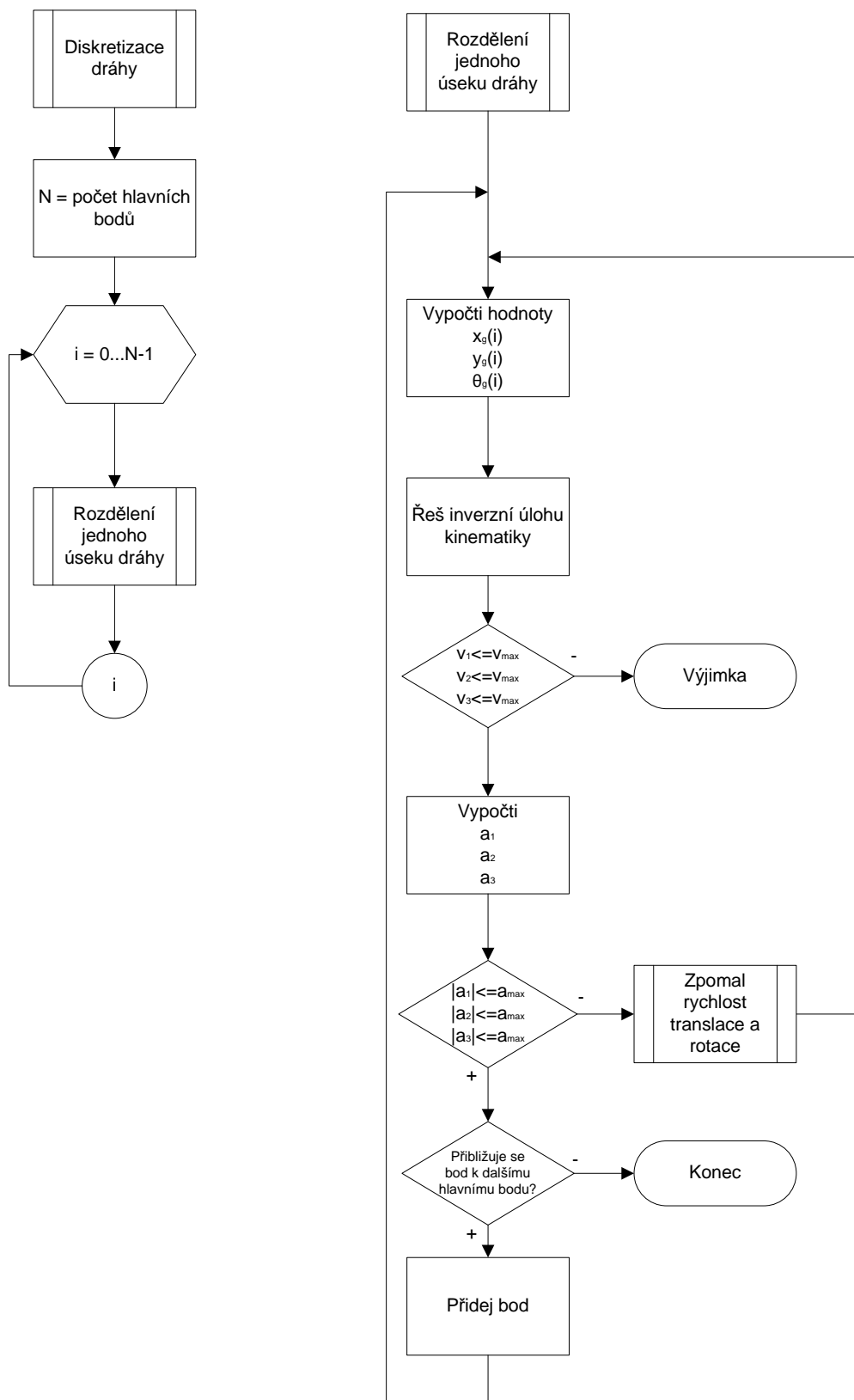
Význam jednotlivých členů je uveden v seznamu zkratk. Z důvodu rozlišení veličin hlavních a diskretizovaných bodů je pro prvně zmíněné použit spodní index i a pro body dráhy je i umístěno v závorkách.

Uvažování rovnoměrně zrychleného pohybu v rovnicích (3.20) až (3.22) je odlišné od rovnic popsanych v literatuře [7], kde je uvažováno se skokovou změnou rychlosti. Domnívám se, že údaje uvedené ve zmíněné literatuře nejsou zcela přesné. Pravdivost mnou uvedených vztahů by měla ověřit simulace.

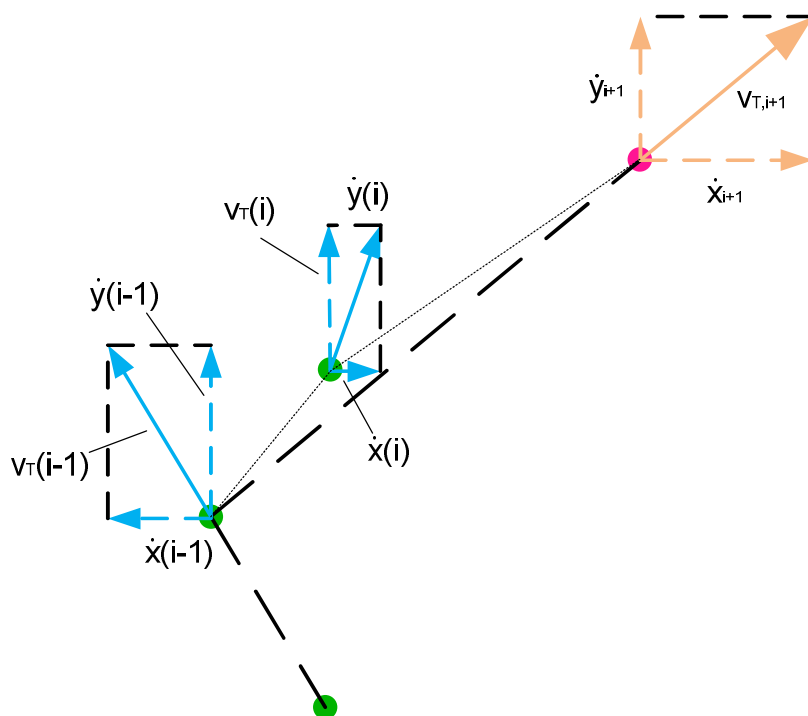
Poloha bodu dráhy na ose x je dána součtem polohy předchozího bodu v Globálním souřadném systému na ose x a dráhou rovnoměrně zrychleného pohybu do dalšího hlavního bodu po dobu diskretizace Δt . Poloha bodu na ose y je dána analogicky. Úhel v bodě dráhy určeném souřadnicemi $(x_G(i), y_G(i))$ je dán součtem úhlu předchozího bodu zadaného v globálních souřadnicích a lineárního přírůstku úhlu v závislosti na vypočtené dráze. Úhel nebyl definován pomocí průměrné úhlové rychlosti rovnoměrně zrychleného pohybu, protože v hlavních bodech dráhy není definováno, jakou úhlovou rychlost má robot mít.

Body vzniklé diskretizací hlavní dráhy budeme označovat jako diskretizované nebo jen jako body dráhy. Zatímco body, ze kterých vznikly, budeme nazývat hlavními body. V programu jsou hlavní body znázorněny tmavě růžovou barvou, zatímco body dráhy jsou menší a jsou znázorněny tmavě zelenou barvou. Rozdíl mezi těmito body můžeme chápat jako kostru dráhy a skutečnou dráhu, která bude robotem absolvována.

Algoritmus vytváření dráhy z hlavních bodů založený na rovnicích (3.20) až (3.22) funguje správně jen v případě, že nechceme zohlednit u každého vytvořeného bodu kinematické vztahy a bylo by nám jedno, jakých zrychlení a koncových rychlostí se kola budou dopouštět. Jestliže chceme tyto údaje zohlednit, musíme po každém vytvoření bodu řešit inverzní úlohu kinematiky a z výsledků zkontrolovat, zda nedošlo k překročení maximálních hodnot rychlostí a zrychlení. Pokud tomu tak bude, je třeba rychlost na dráze zpomalit a opět provést kontrolu. Úvahu nejlépe dokládá vývojový diagram na Obr. 22.



Obr. 22 Vývojový diagram diskretizace dráhy.



Obr. 23 Ukázka zpomalení rychlosti translace robotu v případě překročení maximálního zrychlení.

$$\Delta \dot{x} = \frac{\dot{x}_{i+1} - \dot{x}(i-1)}{20} \quad (3.23)$$

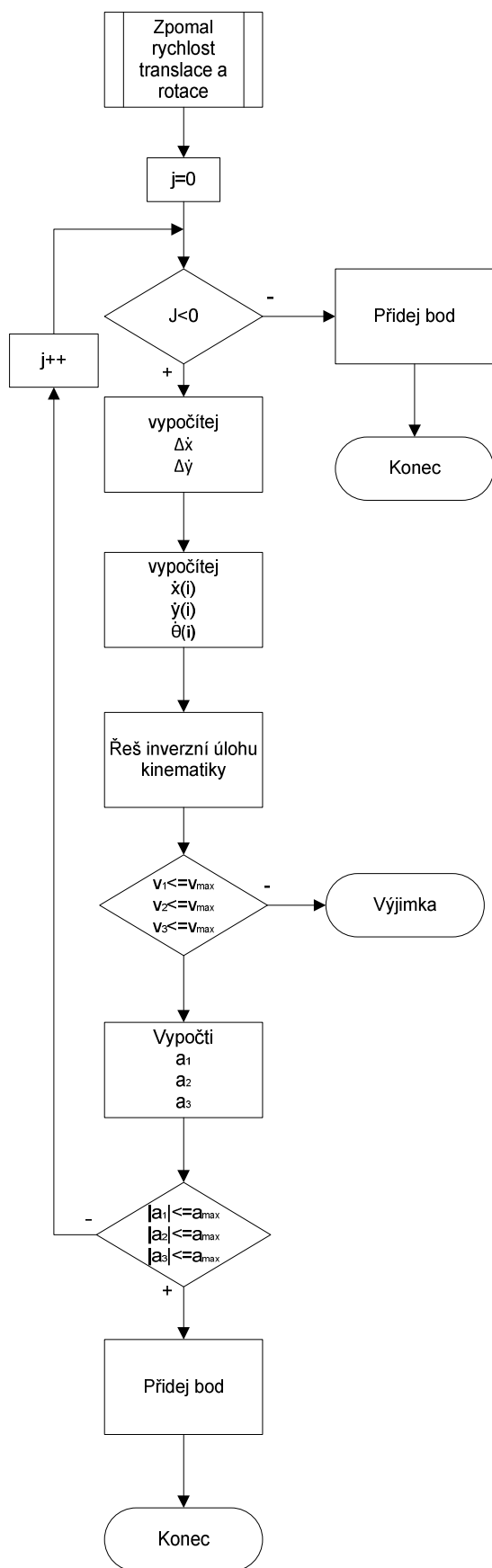
$$\Delta \dot{y} = \frac{\dot{y}_{i+1} - \dot{y}(i-1)}{20} \quad (3.24)$$

$$\dot{x}(i) = \dot{x}(i-1) - \Delta \dot{x} \cdot j \quad (3.25)$$

$$\dot{y}(i) = \dot{y}(i-1) - \Delta \dot{y} \cdot j \quad (3.26)$$

$$\dot{\theta}(i) = \frac{v_T(i)}{v_{T,i+1}} \dot{\theta}(i-1) \quad (3.27)$$

Výše uvedené rovnice platí pro zpomalení translačního a rotačního pohybu robotu, přičemž zpomalení prvně jmenovaného pohybu je dokumentováno i na Obr. 23.

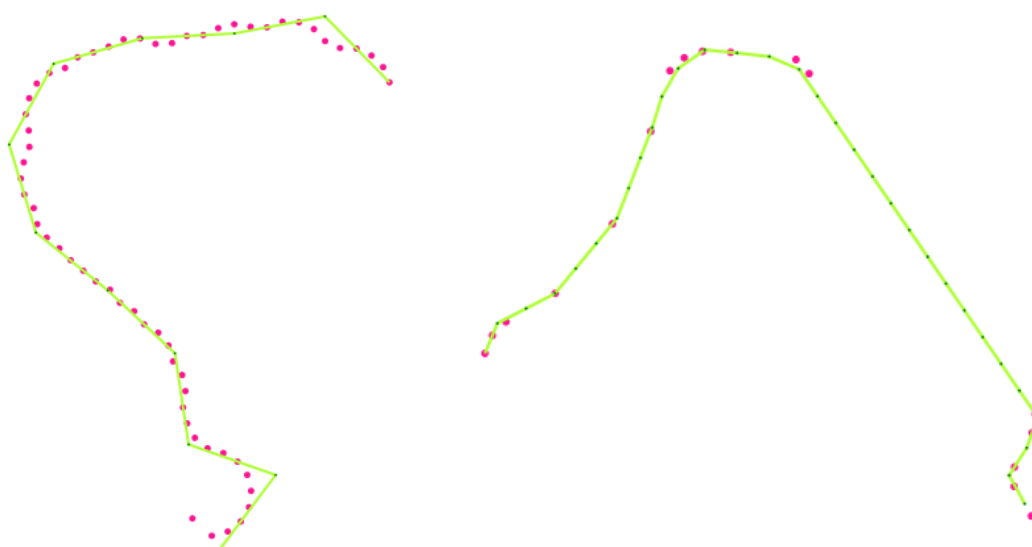


Obr. 24 Vývojový diagram zpomalení translačního a rotačního pohybu.

Čím větší zvolíme hodnotu Δt , tím méně náročné budou výpočty a množství odesílaných dat. Naopak malá hodnota diskretizace vede k přesnějšímu dodržení požadované trajektorie a možnostem navrhovat složitější dráhy. Hodnotu diskretizace nelze ovšem volit libovolně malou, její hodnota je omezena možnostmi komunikace po sběrnici. Jako nejlepší se jevílo volit Δt od hodnoty 0,3 sekundy.

Diskretizovaná dráha nebude nikdy totožná s požadovanou, ale správnou volbou bodů dráhy, rychlostí a rotace všesměrové platformy v jednotlivých úsecích a času diskretizace lze docílit velké podobnosti.

Vývojový diagram uvedený na Obr. 22 lze nazývat také diagramem diskretizačního algoritmu. Ten byl vytvořen za účelem rozdělit dráhu složenou z několika hlavních bodů. Dokáže i při velkých hodnotách Δt a nepravidelném rozložení hlavních bodů velmi dobře požadovanou dráhu aproximovat. Na Obr. 25 jsou ukázány dvě diskretizované dráhy. Růžovou barvou jsou značeny hlavní body dráhy, tmavě zelenou body diskretizované dráhy a světle zelenou pak diskretizovaná dráha.



Obr. 25 Ukázka diskretizace dráhy.

3.6 Relativní a absolutní souřadnice

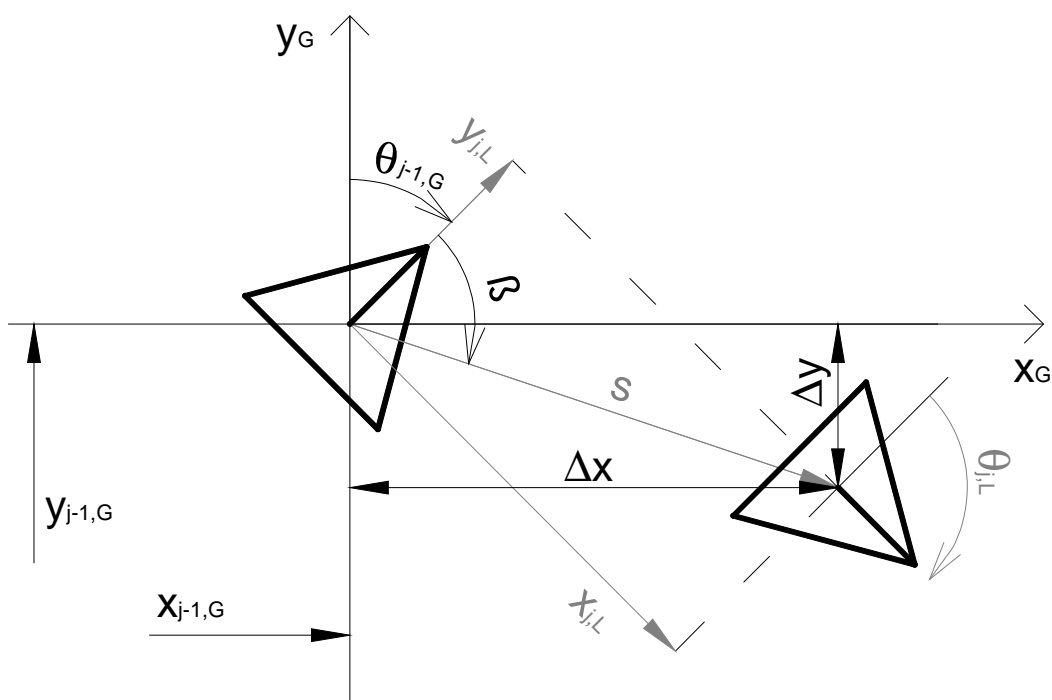
V kapitole 3.1 byly zavedeny dva souřadné systémy a to Globální a Lokální souřadný systém. V Lokálním s. s. definujeme polohy kol a v Globálním polohu těžiště robotu a řešíme zde inverzní úlohu kinematiky.

Z praktického hlediska není příliš vhodné zadávat všechny body jen v absolutních souřadnicích (Globální s. s.). Pokud bychom se rozhodli změnit nějaký úsek dráhy, museli bychom změnit i všechny následující dráhy. Z tohoto důvodu každý bod dráhy má definováno, jestli jeho souřadnice (x, y) a úhel θ jsou zadány relativně nebo absolutně. Můžeme tak vytvářet dráhu složenou jak z absolutních tak z relativních bodů. Pokud je první

bod zadán v relativních souřadnicích, potom je tato souřadnice relativní k počátku Globálního souřadného systému, stejně tak úhel θ je relativní k nulové hodnotě úhlu v Globálním s. s.

Je nezbytné, aby existoval převod mezi oběma souřadnými systémy, neboť před řešením inverzní úlohy kinematiky musí být všechny body převedeny do Globálního s. s.

Mějme zadánu relativní polohu robotu ve stavu j ($x_{L,j}$, $y_{L,j}$, $\theta_{L,j}$). Absolutní polohu robotu zjistíme pomocí dotazu na absolutní polohu robotu v předchozím stavu $j-1$ ($x_{G,j-1}$, $y_{G,j-1}$, $\theta_{G,j-1}$) a přičtením příslušných rozdílů (Δx , Δy , $\theta_{L,j}$).



Obr. 26 Převod mezi souřadnými systémy.

Pro výsledné globální souřadnice dle Obr. 26 platí:

$$x_{G,j} = x_{G,j-1} + s \cdot \sin(\theta_{G,j-1} + \beta) \quad (3.28)$$

$$y_{G,j} = y_{G,j-1} + s \cdot \cos(\theta_{G,j-1} + \beta) \quad (3.29)$$

$$\theta_{G,j} = \theta_{G,j-1} + \theta_{L,j} \quad (3.30)$$

kde pro s a β platí:

$$s = \sqrt{\Delta x^2 + \Delta y^2} = \sqrt{x_{L,j}^2 + y_{L,j}^2} \quad (3.31)$$

$$\tan \beta = \frac{x_{L,j}}{y_{L,j}} \quad (3.32)$$

Programově je hodnota absolutní souřadnice robotu zjištěna z relativní souřadnice pomocí rekurze a procházení obousměrného seznamu.

3.7 Bod dráhy

Bod dráhy je klíčová struktura, respektive třída, která definuje, jak se robot chová v daném bodě dráhy. Nedefinuje jen polohu robotu, jeho rychlost a natočení, ale celou řadu dalších parametrů, které k tomuto stavu vedly. Bez vzájemné provázanosti jednotlivých bodů by nebylo možné vyhodnocovat odlišnost skutečné polohy robotu od žádané a případně na ni reagovat.

Každý bod dráhy má definované atributy viz Tabulka 4.

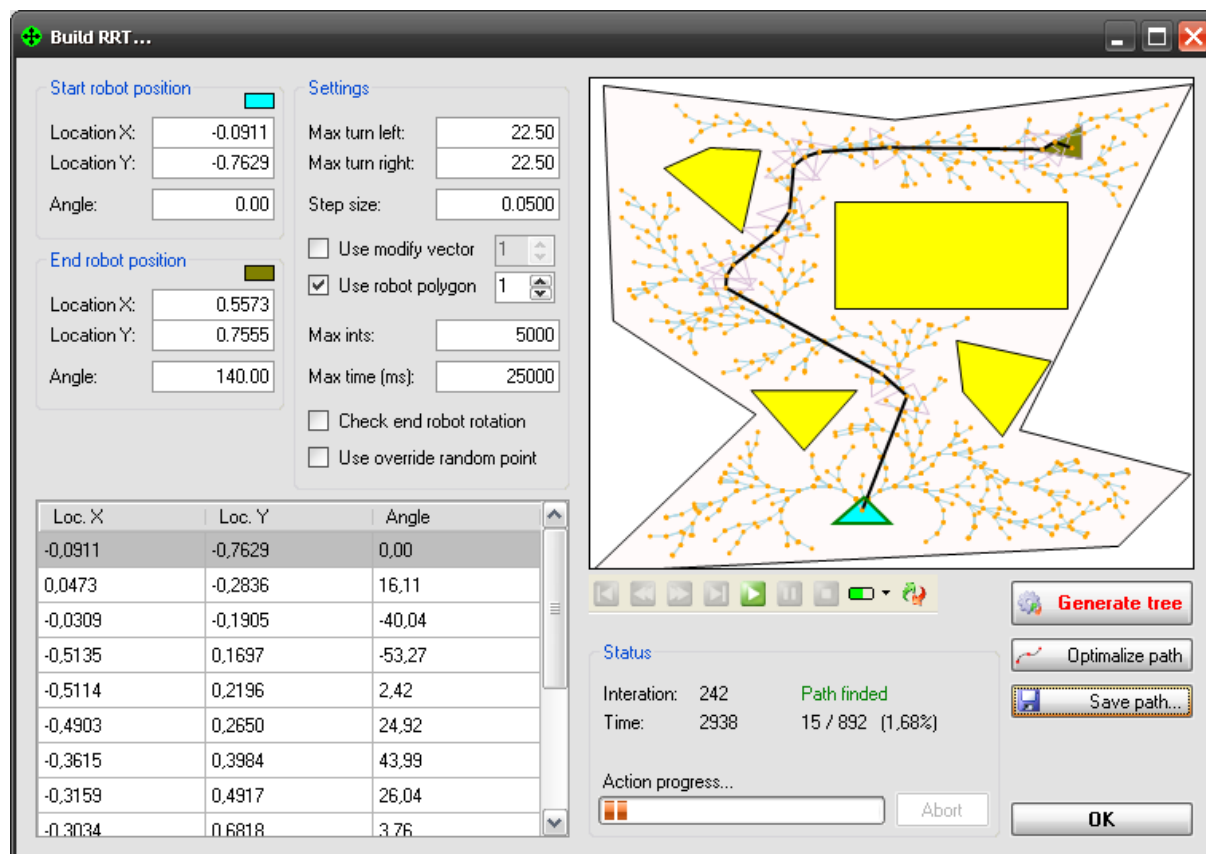
Tabulka 4 Důležité vlastnosti každého bodu dráhy.

Celková dráha do daného bodu
Čas posunu z předchozího do daného bodu
Číslo bodu
Odkaz na další hlavní bod
Odkaz na předchozí bod
Průměrná rychlost mezi předchozím a daným bodem
Rozdíl vektoru dráhy z předchozího do daného bodu promítnutý do osy x_G
Rozdíl vektoru dráhy z předchozího do daného bodu promítnutý do osy y_G
Rozdíl úhlů natočení robotu v dalším hlavním bodě a v daném bodě
Rychlosti všech 3 kol v daném bodě
Souřadnice bodu v ose x Globálního s. s.
Souřadnice bodu v ose y Globálního s. s.
Souřadnice x
Souřadnice y
Typ souřadného systému, ve kterém jsou hodnoty zadány
Úhel v daném bodě a daném souřadném systému
Úhel natočení v globálních souřadnicích
Úhlová rychlost v daném bodě
Vektor dráhy z daného bodu do dalšího hlavního bodu, daný v globálních souřadnicích
Vektor dráhy z předchozího do současného bodu
Vektor rychlosti
Vektor rychlosti do dalšího hlavního bodu
Vektor zrychlení do dalšího hlavního bodu
Velikost dráhy z předchozího bodu po současný
Velikost rychlosti
Zrychlení mezi předchozím a daným bodem

4 Ověření funkčnosti zvoleného řešení

4.1 Příklad dráhy robotu

Pro demonstraci správné funkčnosti navržených algoritmů byla vymodelována dráha v programu Mobile robot studio viz Obr. 27, který byl vytvořen v jedné z předešlých diplomových prací [4], ta byla poté načtena v aplikaci Robot path.

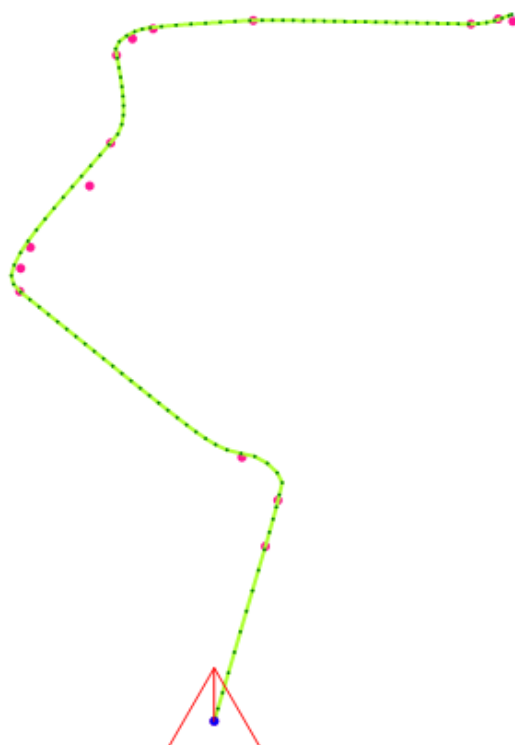


Obr. 27 Vytvoření dráhy v Mobile robot studio [4].

Výsledná dráha viz Obr. 28 po načtení v aplikaci Robot path byla diskretizována dle zadanych parametrů viz Tabulka 5.

Průměr kola	8 cm
Vzdálenost kola od těžiště robotu	18,5 cm
Maximální úhlová rychlost kola	9 ot/s
Čas diskretizace	0,25 s
Maximální zrychlení na obvodu kola	95 cm/s ²

Tabulka 5 Parametry, s nimiž bylo počítáno v aplikaci Robot path, a které tvoří podklad pro níže uvedené grafy.



Obr. 28 Výsledný tvar demonstrační cesty v programu Robot path.

Dráha obsahuje 16 hlavních bodů viz Tabulka 6. Ve zmíněné tabulce jsou uvedeny hodnoty x , y v Globálním souřadném systému, úhel θ , který robot v daném bodě bude mít a rychlost robotu v daném bodě. Bodů dráhy po diskretizaci je 90. Jejich výpis zde není uveden, ale všechny důležité hodnoty byly zaznamenány v grafech. Čas celkové dráhy je roven součinu času diskretizace a počtu bodů ($0,25 \times 90$) a činí 22,5 s. Za tuto dobu by mělo těžiště robotu urazit vzdálenost 6,8 m.

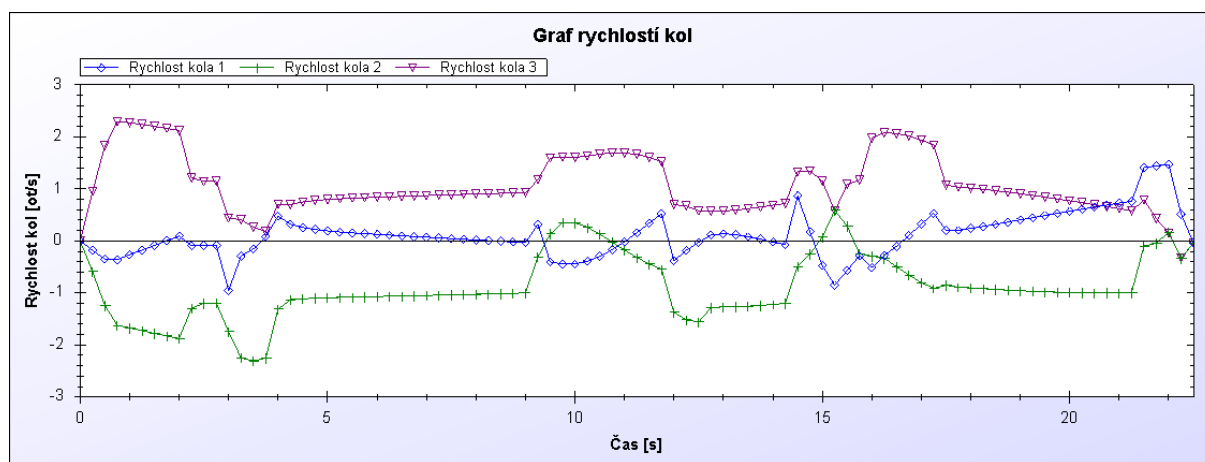
číslo bodu	x_G [cm]	y_G [cm]	úhel θ v daném bodě [°]	rychlost v daném bodě [cm/s]
1	-22,8	-190,7	0	0
2	5	-95,9	16,1	58
3	11,8	-70,9	16,1	34
4	-7,7	-47,6	-40	40
5	-128,4	42,4	-53,3	32
6	-127,9	54,9	2,4	34
7	-122,6	66,3	24,9	30
8	-90,4	99,6	44	30
9	-79	122,9	26	38
10	-75,9	170,5	3,8	26
11	-67,2	179,4	44	28

12	-55,9	184,8	64,4	22
13	-1,6	189,3	85,3	40
14	116,6	187,4	126,9	28
15	131,4	190,2	159,6	22
16	145	194	159,6	0

Tabulka 6 Seznam hlavních bodů dráhy.

Jednotlivé níže uvedené grafy byly pořízeny v programu Robot path jako grafické znázornění vypočtených hodnot v programu. Mohou být zobrazeny pomocí položek v roletě Simulace.

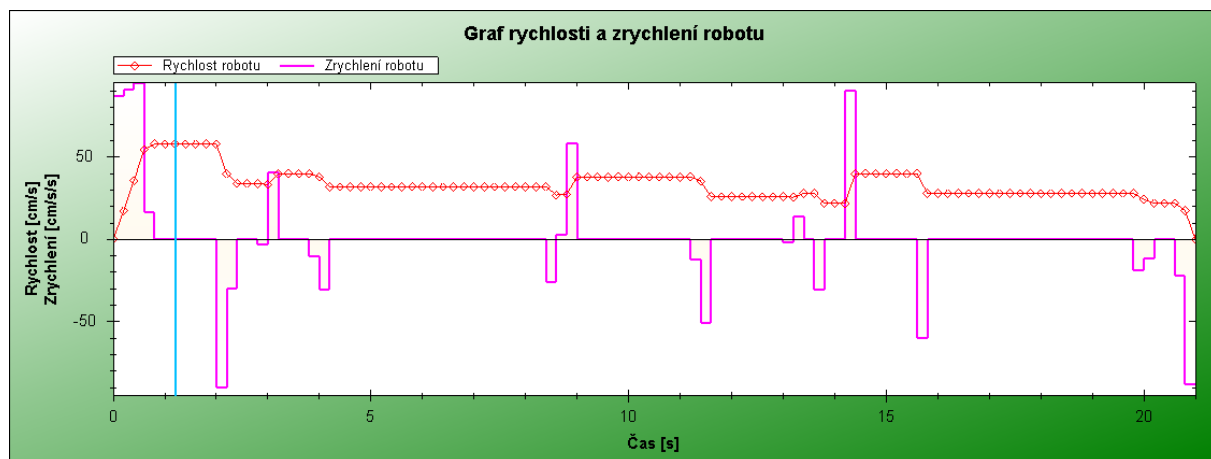
Na Obr. 29 je znázorněna rychlost všech tří kol v závislosti na čase. Je zde vidět, že všechna kola se rozjíždějí z klidové polohy (nulová hodnota rychlosti) a na konci dráhy zpomalují opět na nulovou rychlost. Přechod mezi dvěma hodnotami v grafu nesmí být větší než maximální povolené úhlové zrychlení, jinak by mohlo dojít k prokluzu kol.



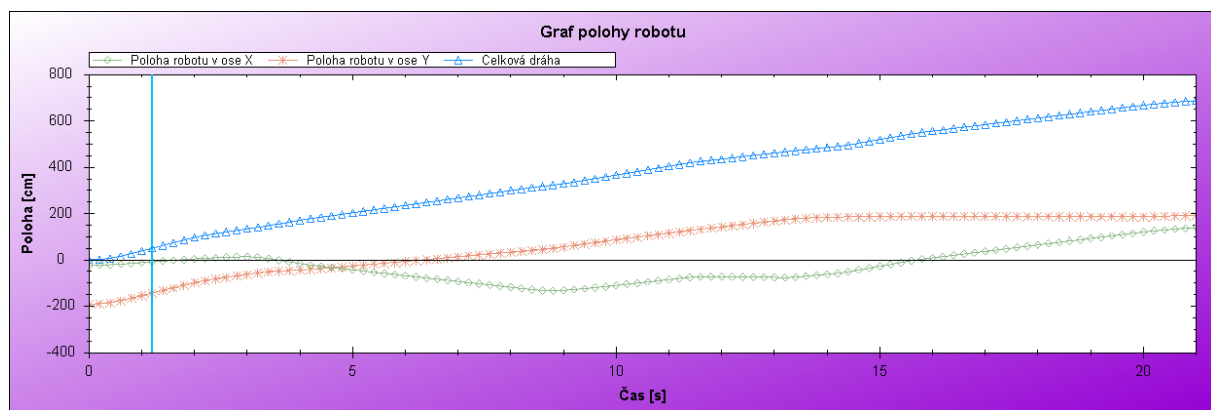
Obr. 29 Graf rychlostí jednotlivých kol v závislosti na čase.

Na Obr. 30 vidíme rychlost těžiště robotu, která opět lineárně roste z nulové hodnoty na začátku a na nulovou hodnotu opět klesá na konci dráhy. Maximální zrychlení těžiště robotu není definováno, ale vychází ze zrychlení kol. Aktuální hodnotu zrychlení pro každý úsek dráhy můžeme také vyčíst ze zmíněného grafu.

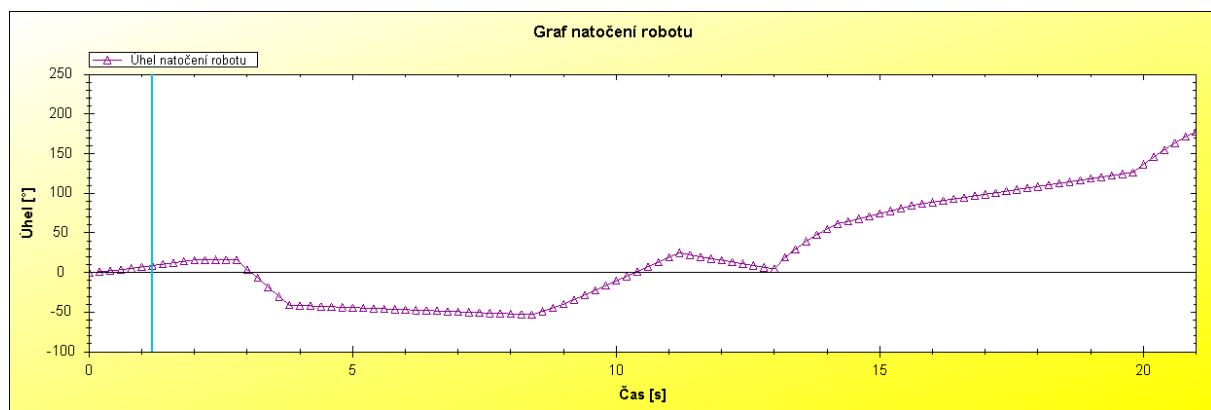
Grafy na Obr. 31 a Obr. 32 znázorňují polohu robotu v Globálním souřadném systému. První z nich ukazuje, jak se mění poloha robotu v ose x, y a celková velikost dráhy. Druhý graf využijeme v případě, že robot provádí jak translační tak rotační pohyb zároveň a tím pádem se mění závislost úhlu θ na čase. Pokud by konal jen translační pohyb, tak by tato hodnota byla po celou dobu konstantní.



Obr. 30 Graf rychlosti a zrychlení robotu v závislosti na čase.

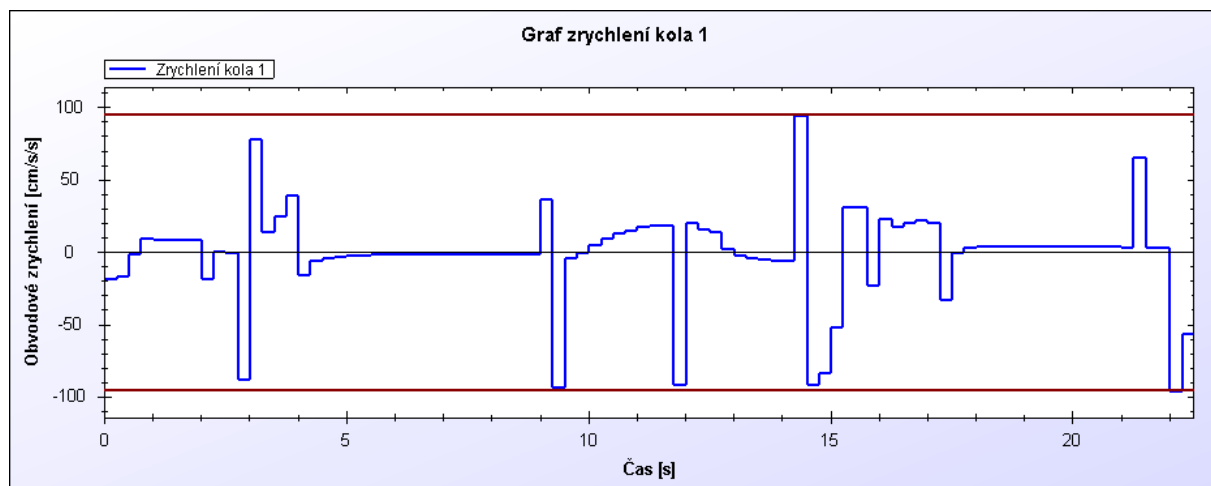


Obr. 31 Graf polohy těžiště robotu v ose x a y a výsledná dráha v závislosti na čase.

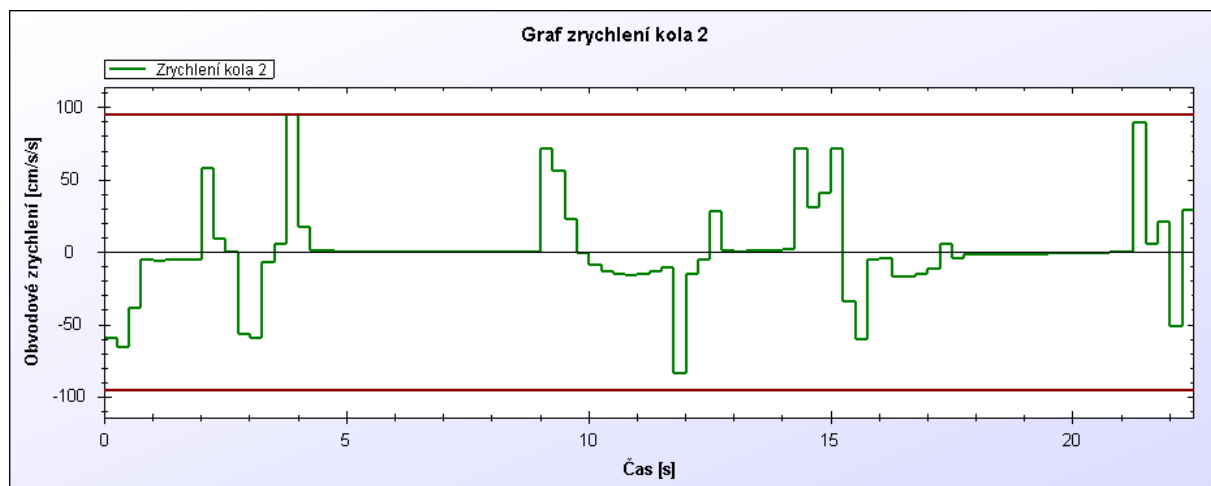


Obr. 32 Graf natočení robotu v závislosti na čase.

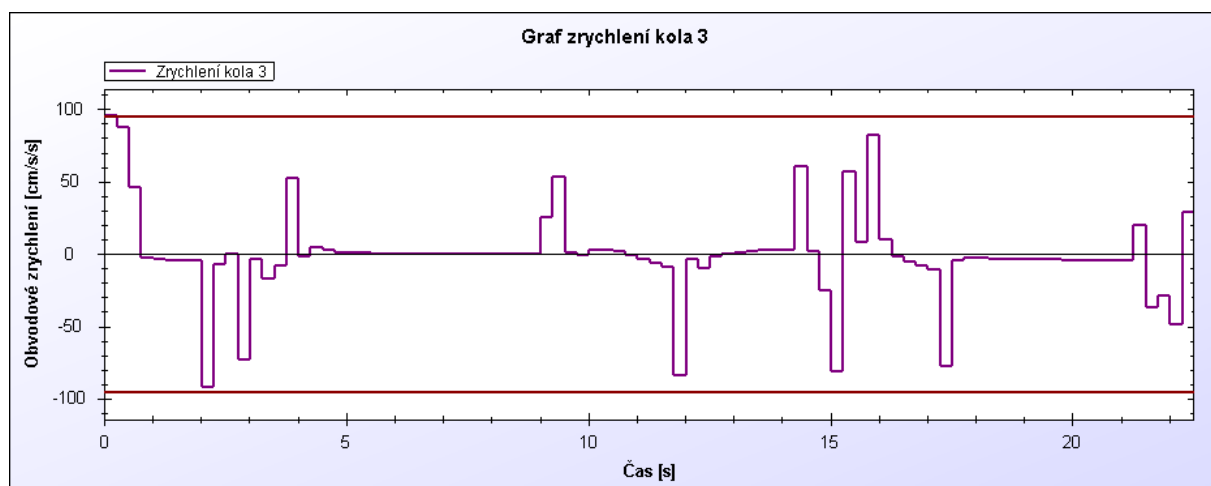
Poslední skupinou grafů jsou grafy zrychlení jednotlivých kol viz Obr. 33 až Obr. 35. Na těchto grafech je znázorněno, jak se mění obvodové zrychlení jednotlivých kol. Tmavě červené čáry znázorňují meze, ve kterých by se zrychlení měla pohybovat, aby nedošlo k prokluzu kol. Grafy dokazují správnost funkce diskretizačního algoritmu, jelikož se všechny hodnoty drží v daném intervalu.



Obr. 33 Graf závislosti zrychlení kola 1 na čase.



Obr. 34 Graf závislosti zrychlení kola 2 na čase.



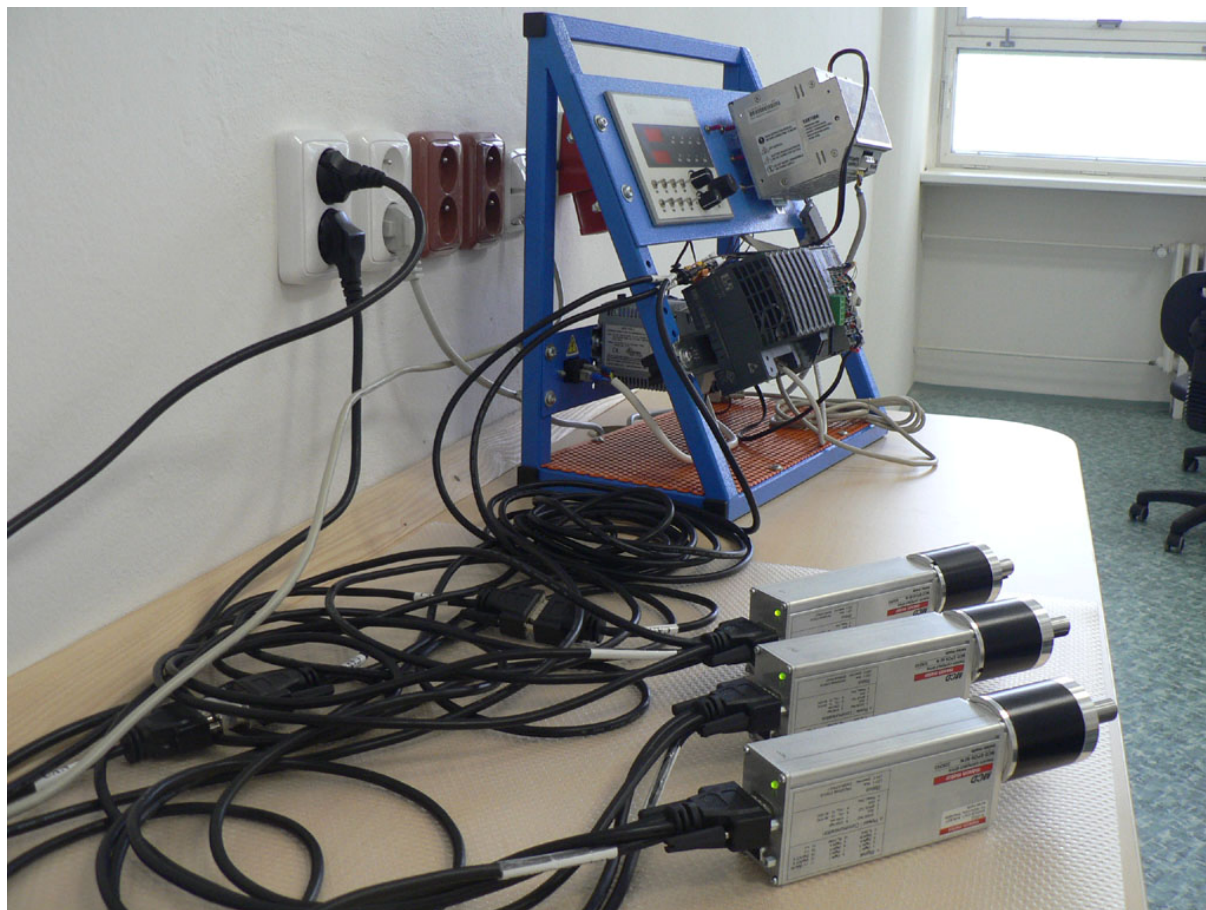
Obr. 35 Graf závislosti zrychlení kola 3 na čase.

Světle modrá čára v grafech označuje jaký bod byl v danou dobu v programu vybrán v editačním poli nebo aktuální pozici robota v čase.

4.2 Test aplikace

Aplikace byla testována v laboratoři odboru aplikované informatiky viz Obr. 36 a byl při ní použit stejnosměrný zdroj s napětím +24VDC. Na motorech byla testována především přesnost dodržení úhlové dráhy ve všech úsecích. V centru pozornosti bylo také testování profilu rychlosti a ověření, zda-li se shoduje s údajem uvedeným v simulačních grafech. Skutečný profil rychlosti byl zjištěn pomocí funkce jednotky EPOS nahrávání dat z enkodérů.

Zjištění celkové přesnosti dráhy kol bylo bráno jako rozdíl mezi celkovou požadovanou úhlovou dráhou v celém úseku dráhy a skutečnou úhlovou dráhou, kterou motor opsal.



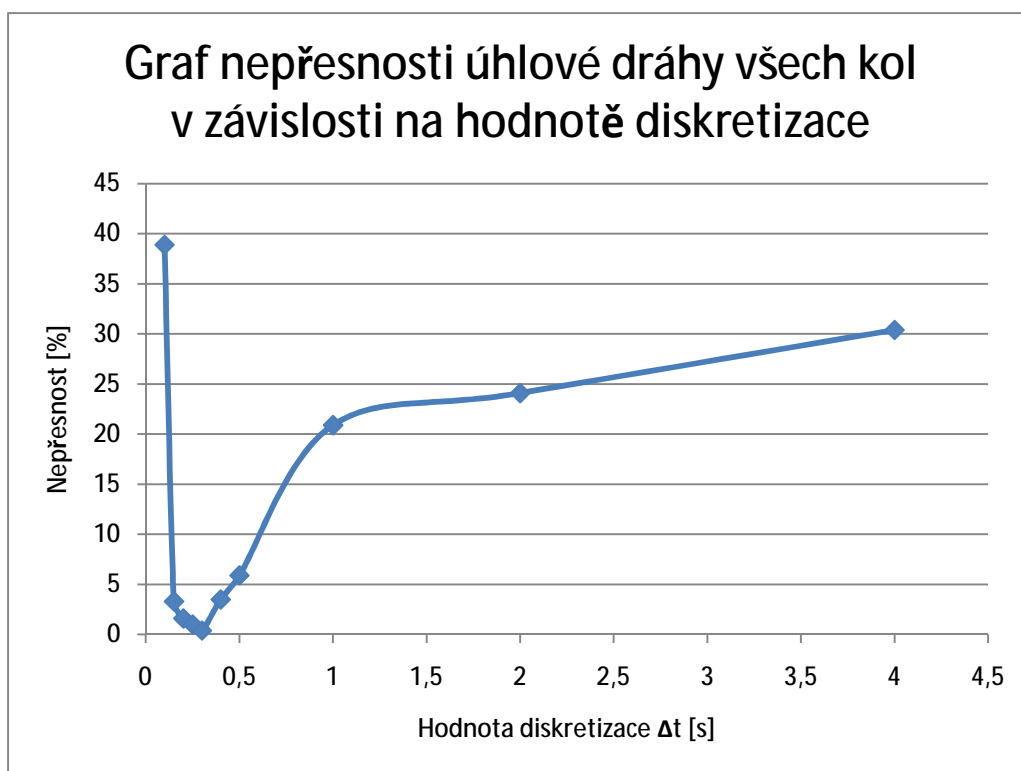
Obr. 36 Testování funkčnosti aplikace v laboratoři odboru aplikované informatiky.

Přesnost řízení pro demonstrační dráhu uvedenou v kapitole 4.1 byla 1% viz Tabulka 7.

Tabulka 7 Ověřená přesnost úhlové dráhy kol v demonstrační dráze.

Celková doba absolvování dráhy	22,50 s
Předpokládaná celková doba absolvování dráhy	22,10 s
Úhlová dráha kola 1	6,55 ot
Předpokládaná úhlová dráha kola 1	6,50 ot
Úhlová dráha kola 2	20,74 ot
Předpokládaná úhlová dráha kola 2	20,74 ot
Úhlová dráha kola 3	24,08 ot
Předpokládaná úhlová dráha kola 3	23,62 ot
Nepřesnost úhlové dráhy kola 1	0,88%
Nepřesnost úhlové dráhy kola 2	0,02%
Nepřesnost úhlové dráhy kola 3	1,97%
Celková nepřesnost úhlové dráhy	1,00%

Na grafu viz Obr. 37 je zachycena přesnost dráhy v závislosti na volbě hodnoty diskretizace pro případ demonstrační dráhy.



Obr. 37 Graf nepřesnosti úhlové dráhy všech kol v závislosti na hodnotě diskretizace.

Z grafu na Obr. 37 nelze usuzovat, že zvýšení hodnoty diskretizace vede k větší nepřesnosti, jelikož pro méně členité dráhy a pro dráhy, v nichž nedochází ke kombinaci rotačního a translačního pohybu, může být dostačující. Nepřesnost hodnot Δt menší než 0,2s je způsobena rychlostí komunikace a rychlostí řídicí jednotky EPOS, která není schopna takto

rychlé povely o změně rychlosti přenést. Z těchto závěrů vyplývá, že nejlepší volbou času diskretizace jsou 0,3 s, při kterých je dosaženo nejnížší nepřesnosti (pod 1%).

5 Závěr

V úvodních kapitolách jsem zaměřil svou pozornost na problematiku spojenou s všesměrovými podvozky, jejich výhodami a nevýhodami pro jednotlivá použití. Nebyla opomenuta ani rozdílnost oproti klasickým podvozkům. Základním stavebním kamenem pro praktickou část byly inteligentní pohony od firmy Maxon, jež se staly alfou a omegou veškerých úvah v oblasti řízení. Prvním krokem praktického návrhu řízení bylo sestavení kinematických rovnic. Kinematické vztahy byly vytvořeny pro pohyb složený z translační i rotační složky. Tento fakt je nespornou výhodou, jelikož umožňuje rozmanitý pohyb robotu. Jeho stinnou stránkou je potřeba úpravy navržené dráhy do podoby složené z několika stejných časových úseků, ve kterých budou kinematické vztahy řešeny (tzv. diskretizace). Podmínkou je, že výsledky kinematických vztahů musí být v takové formě, aby splňovaly kritéria daná v podobě maximálního zrychlení a maximální rychlosti kol. Pokud by tato kritéria nebyla dodržena, výsledná přesnost dráhy robotu by značně utrpěla díky prokluzu kol. V této diplomové práci se podařilo navrhnout takový model, aby tyto podmínky byly splněny. Základem úvah byl rovnoměrně zrychlený pohyb, jenž bylo možné dobře aplikovat díky kvalitním pohonům. Ty neposkytovaly jen kvalitní možnost řízení, ale i možnosti zpětné vazby v podobě přesných údajů z inkrementálních snímačů polohy. Díky těmto údajům byla ověřena skutečně absolvovaná dráha kol. Jediným nedostatkem v ověřování byla absence nově vyvíjeného podvozku, jenž by nesporně výsledný model uvedl v činnost.

LITERATURA

- [1] SOUKUP, K. *Řízení podvozku všesměrového mobilního robotu*. Disertační práce. Vysoké Učení Technické v Brně. Fakulta Strojního Inženýrství. 2004. 61 stran. Školitel Branislav Lacko.
- [2] KÁRNÍK, L. Modelování a analýza servisních robotů. [online]. [cit. 14. dubna 2010]. <http://robot2.vsb.cz/elekskripta/servisni_roboty/interest131222111111111111.htm>
- [3] Maxon motor. *Intelligent compact drivers*. [online]. duben 2008. [cit. 15. dubna 2010]. <http://www.maxonmotor.com/downloads/MCD_2008_e.pdf>
- [4] KOCH, Z. *Počítačová simulace pohybu a plánování trajektorie mobilního robotu v pseudo 3D prostředí*. Diplomová práce. Vysoké Učení Technické v Brně. Fakulta Strojního Inženýrství. 2008. 43 stran. Vedoucí Radomil Matoušek.
- [5] BAEDE, T. *Motion control of an omnidirectional mobile robot*. Zpráva ze stáže. National University of Singapore. Faculty of Engineering. 18. září 2006. 50 stran.
- [6] R.P.A. van HAENDEL. *Design of an omnidirectional universal mobile platform*. Zpráva ze stáže. National University of Singapore. Faculty of Engineering. září 2005. 70 stran.
- [7] CHIA-JU, W. *Kinematic Control of Omni-directional Robots for Time-optimal Movement between Two Configurations*. National Yunlin University of Science and Technology. Graduate School of Engineering Science and Technology. 2007.
- [8] BROŽ, V. *Programovatelný regulátor rychlosti a polohy pro sběrnici CAN*. [online]. [cit. 22. dubna 2010]. <http://www.odbornecasopisy.cz/index.php?id_document=30895>
- [9] BROŽ, V. *Nové možnosti pohonů Maxon*. [online]. [cit. 22. dubna 2010]. <http://www.odbornecasopisy.cz/index.php?id_document=34023>
- [10] *Jednotky řízení polohy EPOS*. UZIMEX PRAHA, spol. s r.o. 2009. [online]. [cit. 8. dubna 2010]. <http://www.uzimex.cz/download.php?file=/soubory/20050608_epos_pruvodce_1.2.pdf>
- [11] Codesion, cohesion in code control. [online]. Hosting verzovacího systému Subversion. <<http://codesion.com>>
- [12] KUČEROVÁ, H. *Diagram tříd*. [online]. 31. 3. 2007 [cit. 15. dubna 2010]. Praha: Vyšší odborná škola informačních služeb. <<http://web.sks.cz/users/ku/pri/tridy.htm>>

- [13] Planetové převodovky. UZIMEX PRAHA, spol. s r.o. 2009. [online]. [cit. 10. dubna 2010]. <<http://www.uzimex.cz/Vyrobce/maxon-motor-ag/Prevodovky/Planetove-prevodovky.html>>
- [14] Vacek, F. *CANopen – vyšší komunikační protokol pro vestavné sítě*. Překlad z Holger Zeltwanger, CAN in Automation. <http://www.odbornecasopisy.cz/index.php?id_document=32279>
- [15] KOCIAN, T. *Dálkové řízení mobilních robotů*. Diplomová práce. 2002. 55 s. Vysoké Učení Technické v Brně. Fakulta Strojního Inženýrství.

SEZNAM PŘÍLOH

Příloha č. 1 – Přiložené CD-R médium s obsahem

- elektronická podoba této diplomové práce(\text\DP_Olsa.pdf)
- spustitelný software: Aplikace Robot path
- e-dokumentace knihoven